

DTIC FILE COPY

2

AD-A232 158

STRATEGY COMBINING REGIONALLY  
ADAPTED PROCESSES (SCRAP)  
FINAL REPORT

D'

DTIC  
ELECTE  
FEB 26 1991  
S B D

GENERAL  
SERVICES  
INCORPORATED

111 South Bedford Street, Burlington, Massachusetts 01803 (617) 273-4770

91 2 19 006

2

SSI-TR-183

STRATEGY COMBINING REGIONALLY  
ADAPTED PROCESSES (SCRAP)  
FINAL REPORT

DTIC  
ELECTE  
FEB 26 1991  
S B D

Prepared by:

M. M. Pervaiz  
Spectral Sciences, Inc.  
99 South Bedford Street, #7  
Burlington, MA 01803-5169

Prepared for:

U. S. Army Research Office  
P.O. Box 12211  
Research Triangle Park, NC 27709-2211

Under Contract No. DAAL03-90-C-0014

January 1991

**DISTRIBUTION STATEMENT A**

Approved for public release  
Distribution Unlimited

# REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE 30 January 1991	3. REPORT TYPE AND DATES COVERED Final Report - 01JUL90 - 31DEC90	
4. TITLE AND SUBTITLE Strategy Combining Regionally Adapted Processes (SCRAP)			5. FUNDING NUMBERS  DAAL03-90-C-0014	
6. AUTHOR(S) M. M. Pervaiz				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Spectral Sciences, Inc. 99 South Bedford Street, #7 Burlington, MA 01803-5169			8. PERFORMING ORGANIZATION REPORT NUMBER  SSI-TR-183	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) U. S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211			10. SPONSORING/MONITORING AGENCY REPORT NUMBER  ARO 27961.1-E6-S&I	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) <p>A local Navier-Stokes (NS) algorithm to simulate a physically and geometrically complex flow is presented. The domain is subdivided into blocks which facilitate the generation of global mesh and splitting of schemes. Basic types are <i>inviscid</i>, <i>viscous</i> and <i>interaction</i> blocks. In the interaction block a local procedure is used to decide whether to use NS or Euler equations. Each block also employs subdomain adaptation to capture small scale features.</p> <p>The method concentrates resources by modeling complex physics in only important regions. Physics adaptation treats those cells for which local stress is more than 1% of the maximum value as viscous cells. Subdomain adaptation is carried out by monitoring the first differences of selected variables. The cells for which these differences are large are locally divided.</p> <p>The SCRAP technique is demonstrated via a 2-D model problem. It significantly reduces computational requirements and is comparatively more accurate than conventional schemes. Up to two orders of magnitude savings in computer time have been achieved, compared to a globally fine solution of full NS equations. The agreement with previous full NS computations over fine grids is excellent. Future work will extend the approach to 3-D configurations. Phase II will simulate flows for complex configurations at large attack angles.</p>				
14. SUBJECT TERMS Navier-Stokes      Interaction Blocks      Subdomain Adaptation Physics Adaptation			15. NUMBER OF PAGES 53	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

# TABLE OF CONTENTS

SECTION	PAGE
NOMENCLATURE . . . . .	vi
1 INTRODUCTION . . . . .	1
1.1 Problem Overview . . . . .	1
1.2 Overview of Phase I Accomplishments . . . . .	2
1.3 Overview of Report . . . . .	4
2 PHASE I TECHNICAL OBJECTIVES . . . . .	5
3 NUMERICAL DETAILS . . . . .	7
3.1 Governing Equations . . . . .	8
3.2 Normalization of Governing Equations . . . . .	10
3.3 Integration Scheme . . . . .	12
3.3.1 Inviscid Terms . . . . .	12
3.3.2 Viscous Terms . . . . .	14
3.4 Boundary Conditions . . . . .	18
3.4.1 Free-Slip Rigid Walls . . . . .	18
3.4.2 No-Slip Rigid Wall . . . . .	20
3.4.3 Inflow/Outflow Boundary Conditions . . . . .	20
3.5 Stability Analysis . . . . .	27
3.5.1 One-dimensional Convection-diffusion Model . . . . .	27
3.5.2 Time Steps in 2-D . . . . .	30
3.5.3 Actual Implementation of Time Steps . . . . .	30
3.6 Interactive Block Grid Generator . . . . .	31
3.7 Physics Adaptation . . . . .	35
3.8 Subdomain Adaptation . . . . .	38
4 PHASE I NUMERICAL RESULTS . . . . .	39
4.1 Model Problem . . . . .	39
4.2 Euler Solution . . . . .	41
4.3 Full Navier-Stokes Solution . . . . .	41
4.4 Physics Adaptation Solution . . . . .	43
4.5 Physics and Subdomain Adaptation Solution . . . . .	44
4.6 CPU Time Comparisons . . . . .	46
4.7 Adapted Solution with Alternate Outflow Boundary Condition . . . . .	47
5 CONCLUSIONS AND SUMMARY . . . . .	50
6 REFERENCES . . . . .	52



Session For	
MS GRA&I	<input checked="" type="checkbox"/>
MC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

## LIST OF FIGURES

FIGURE	PAGE
1    An Arbitrary Cell C with Nodes i,j,k,l . . . . .	13
2    Staggered Cells for the Computation of Edge Gradients . .	17
3    Physical Cells and their Images Adjacent to a Wall . . . .	18
4    Coordinate Systems for Inflow/Outflow Boundaries . . . . .	22
5    User Selected Node Points for 14 Blocks in a Computational Domain of an 8% Circular Arc Cascade with a Flap . . . . .	32
6    User Selected Numerical Schemes for the 14 Blocks in the Computational Domain of an 8% Circular Arc Cascade with a Flap . . . . .	33
7    Initial Interior Grid Assembly for the Computational Domain of an 8% Circular Arc Cascade with a Flap . . . . .	34
8    Grid Assembly Produced by an Elliptic Grid Generator for the Computational Domain of an 8% Circular Arc Cascade with a Flap . . . . .	35
9    Geometry and Initial Grid (51x173) for Supersonic Flow Over an 8% Circular Arc Cascade . . . . .	40
10   Density Contours for the Euler Solution for 8% Circular Arc Cascade on a 51x173 Mesh System . . . . .	41
11   Density Contours for the Full Navier-Stokes Solution for 8% Circular Arc Cascade on a 51x173 Mesh System . . . .	42
12   Domain Splitting for Physics Adaptation Solution for 8% Circular Arc Cascade on a Composite Mesh of 51x173 Points . . . . .	43
13   Density Contours for the Physics Adapted Solution for 8% Circular Arc Cascade on a 51x173 Initial Grid . . . . .	44
14   (a) Final Spatially Adapted Grid Obtained by Utilizing SCRAP Technique, and (b) Density Contours on this Grid with an Increment of 0.05 . . . . .	45
15   CPU Time Comparisons for Various Approaches . . . . .	46
16   (a) Final Spatially Adapted Grid Obtained by Utilizing SCRAP Technique, and (b) Density Contours on this Grid with an Increment of 0.05 . . . . .	48

## LIST OF FIGURES

FIGURE		PAGE
17	Mach Number Contours for the Fully Adapted Solution for an 8% Circular Arc Cascade . . . . .	48
18	Pressure Distribution Along the Lower Channel Wall for the 8% Circular Arc Cascade . . . . .	49

## NOMENCLATURE

$a$	speed of sound
$a$	adiabatic or isothermal wall parameter, Subsection 3.4
$A$	cell area
$c$	constant wave speed for stability analysis
$C_p$	specific heat at constant pressure
$D$	a cell dimension defined by Eq. (103)
$F$	a factor (Eq. (105)) that keeps the inviscid and viscous time steps to within the same order of magnitude
$\vec{F}_i$	inviscid flux vector for x-coordinate
$\vec{F}_v$	viscous flux vector for x-coordinate
$\vec{G}_i$	inviscid flux vector for y-coordinate
$\vec{G}_v$	viscous flux vector for y-coordinate
$\vec{H}_i$	inviscid flux vector for z-coordinate
$\vec{H}_v$	viscous flux vector for z-coordinate
$k$	thermal conductivity
$L$	matrix of the eigenvectors associated with the Jacobian $F_U$
$M$	Mach number
$p$	pressure
$Pr$	Prandlt number
$\vec{q}$	heat flux vector
$Q$	characteristic variable vector with components $(q_1, q_2, q_3, q_4)$
$Re_c$	cell Reynolds number $\geq 2$
$Re$	cell Reynolds number
$R_{\pm}$	Riemann invariants
$S$	constant in Sutherland law
$t$	time coordinate
$T$	temperature
$(u, v, w)$	velocity components along $(x, y, z)$ coordinate directions

$\bar{U}$	state vector, conservative vector
$v$	speed, magnitude of total velocity = $\sqrt{u^2+v^2+w^2}$
$(x,y,z)$	spatial coordinates
$\alpha$	angle of attack
$\alpha$	local angle of a solid wall with x-axis, Subsection 3.4
$\alpha$	coefficient for viscous changes, Subsection 3.3
$\beta$	coefficient for viscous changes, Subsection 3.3
$\gamma$	ratio of specific heats, typically 1.4
$\Gamma$	CFL number $\leq 1$
$\partial C$	boundary (or perimeter) of a cell C
$\delta_{ij}$	Kronecker delta function
$\epsilon$	total energy per unit volume
$\lambda$	second coefficient of viscosity
$\mu$	first coefficient of viscosity, dynamic viscosity
$\nu$	kinetic viscosity
$\eta$	direction along the inflow/outflow boundary
$\Lambda$	diagonal matrix with elements as eigenvalues of Jacobian $F_U$
$\xi$	direction normal to inflow/outflow boundary surface
$(\xi,\eta)$	generalized curvilinear coordinates
$\rho$	density
$\Sigma$	cell diffusion number $\leq \frac{1}{2}$
$\bar{\tau}$	stress tensor

#### Subscripts and Superscripts

$c$	corrected value, Subsection 3.4
$c$	frozen or constant value, Subsection 3.4
$i$	inviscid
$m$	maximum



n	time level
p	predicted value
r	reference
t	tangential component (along solid wall)
U	Jacobian matrix w.r.t. state vector
v	viscous
w	wall
$\eta$	a surface component for inflow/outflow boundary
$\xi$	component normal to the inflow/outflow boundary
*	non-dimensional form
$\infty$	free stream

#### Abbreviations and Acronyms

ARO	Army Research Office CFD Computational Fluid Dynamics
CFL	Courant-Friedrichs-Lewy
CNS	Complete Navier-Stokes
CPU	Central Processing Unit
IBGG	Interactive Block Grid Generator
NS	Navier-Stokes
PI	Principal Investigator PNS Parabolized Navier-Stokes
RMS	Root Mean Square
SCRAP	Strategy Combining Regionally Adapted Processes
SSI	Spectral Sciences, Inc.
TLNS	Thin-Layer Navier-Stokes

## 1. INTRODUCTION

### 1.1 Problem Overview

The capability of carrying out numerical computations of flow fields around finned missile configurations is of significant importance to the U.S. Army.<sup>(1-2)</sup> Spatial marching techniques have previously been applied to predict three-dimensional viscous flow about axisymmetric bodies and finned geometries of interest to the Army for low angles of attack.<sup>(3-7)</sup> For such a technique the solution is marched spatially down the body in the predominant flow direction. However, this technique is only applicable to problems for which the flow is supersonic and does not contain imbedded subsonic regions or regions where the flow separates in the marching direction. A time marching subdomain adaptive technique that utilizes Navier-Stokes (NS) equations for separated regions or in the vicinity of solid surfaces, and Euler equations elsewhere, was proposed to overcome the shortcomings of spatial marching techniques.<sup>(8)</sup> This technique was titled Strategy Combining Regionally Adapted Processes (SCRAP).

The overall stated objective of the research is to carry out 3-D numerical simulations of finned missiles and guided projectiles which have a gap between the inner fin edge and the body surface. The simulation of the flow field about these configurations at a non-zero angle of attack is important from the standpoint of understanding the physical phenomena and predicting the magnitude of the forces acting on the fins and the projectiles. The flow field simulation of the full configuration (rather than just the fins and gap regions) is necessary because the flow field associated with the body affects the one associated with the fins and vice-versa at high angles of attack. The flow fields are geometrically complicated because of the 3-D configurations and the close proximity of fins and body. This makes the process of initial grid generation complicated. Also, the fin/body configurations have complex physics associated with them. This may include shock waves, boundary layers, recirculation regions, etc., and their interactions. It is important to consider numerical schemes that have good shock-capturing properties. Full NS computation with an appropriate

turbulence model may be prohibitively expensive. However, full NS solutions is only needed in the gap regions and in the boundary layers or recirculation regions. In addition, it may also be needed in the regions of interactions of features like shock waves and boundary layers. For much of the computational domain, Euler solution is adequate. Therefore, it makes sense to consider different numerical schemes in different selected regions to speed up the convergence of the calculation.

Although the SCRAP technique was originally proposed to simulate transonic or supersonic flow about finned missiles and guided projectiles, this methodology can be applied to any physically and/or geometrically complex flow field. The issue of the complexity of the grid generation can be partially simplified by considering a *multiblock* gridding technique. In this approach each component of the overall configuration has an independent grid structure which is called a *block*. The multiblock SCRAP technique simplifies physical complexity by associating a numerical scheme to each of the blocks. Thus, one can have viscous blocks where NS equations are solved and *inviscid* blocks where one carries out the solution to Euler equations. Such a scheme-splitting technique can efficiently model complex physics if the domains of viscous and inviscid regions are known a priori. The SCRAP technique addresses this issue by considering *interaction* blocks where a decision is locally made whether to use Euler or NS equations. This dynamic physics switching is termed here as *physics (or equation) adaptation*. The interaction blocks will be important for the cases where the viscous/inviscid interaction region develops as part of the numerical solution. The SCRAP technique also implements *subdomain adaptation* in which the algorithm automatically implements additional spatial resolution by locally dividing cells in the regions of enhanced gradients of selected flow field variables.

## 1.2 Overview of Phase I Accomplishments

This subsection summarizes the key accomplishments of Phase I. These accomplishments pertain to the implementation of the overall SCRAP technique in two spatial dimensions. Extension to 3-D will be carried out in Phase II. The accomplishments include:

- Development of an Interactive Block Grid Generator (IBGG).
- Development of a general purpose code that implements physics and subdomain adaptation on multiblock grids.
- Demonstration of the potential of SCRAP technique for a model problem.

For the Interactive Block Grid Generator (IBGG) the flow domain is divided into blocks, and each of these blocks is associated with a given numerical scheme. These blocks are classified as Euler, Navier-Stokes, Thin-Layer Navier-Stokes (TLNS), or *interaction*, depending upon the scheme being utilized. Interaction blocks are the ones in which a local decision is made for each of its cells whether to use an Euler or NS solver. In other words physics adaptation is carried out on these interaction blocks. The user specifies the nodes of the boundaries of the blocks which must have node-to-node matching at inter-block interfaces. Utilizing the interblock boundary point information the interior grid of each of the blocks is generated. Subsequent to that, all the nodes that are multiply defined are removed and the overall boundary nodes are adjusted. Then, all the cells with the same numerical algorithm are stored together in the computer memory; these cells need not be physically contiguous. Cells in each block can be subsequently spatially divided by the algorithm to incorporate resolution adaptation. Regions of cells can also be initially manually subdivided to incorporate additional resolution or pre-embedding. Various stages of the grid assembly can be graphically viewed and debugged interactively in this IBGG. For example, nodes can be moved individually or in concert with an elliptic grid generator. Similarly the information about the connectivity of various elements (cells, nodes, boundary points) can be queried interactively.

The overall algorithm is decoupled into a viscous and inviscid contribution. This means that Euler equations are solved everywhere, including the viscous region. This is followed by the determination of viscous cells in the interaction blocks. These cells are lumped together with cells in the viscous regions. Additional viscous contributions (corresponding to NS or TLNS equations) are computed for these viscous cells and added to the inviscid contributions. The decoupled nature of the algorithm allows different time steps to be taken for viscous and inviscid contributions. This

is in line with the concept of local time stepping for carrying out solutions to steady state problems. Physics adaptation is incorporated by finding the maximum shear stress in the cells of viscous and interaction blocks. Those cells of the interaction block for which the shear stress is more 1% of the maximum shear stress are regarded as viscous cells. Subdomain adaptation is carried out by examining the gradients of selected flow variables in the domains. The cells for which the first differences of these variables exceed a threshold limit are locally subdivided.

The potential of the SCRAP technique is demonstrated for a supersonic flow over an 8% circular arc cascade. This demonstration clearly establishes that simultaneous physics and resolution adaptation works for the sample problem. The technique provides significant savings in CPU time and storage over conventional structured grid algorithms employing a full NS solver. An increase of about two orders of magnitude in computational speed was achieved when compared to a conventional full NS solution over a fine grid of the same mesh size as the smallest grid size chosen by the spatial resolution adaptation.

### 1.3 Overview of Report

The following section summarizes the technical objectives of Phase I and the status of accomplishments. Section 3 describes the details of the methods utilized to achieve each part of the technical objectives. Section 4 describes the actual numerical results obtained by utilizing the SCRAP technique. Conclusions of this report are summarized in Section 5.

## 2. PHASE I TECHNICAL OBJECTIVES

The stated specific objectives of the Phase I research<sup>8</sup> were:

1. Implement integration schemes for solving Euler, thin-layer Navier-Stokes (TLNS) and complete Navier-Stokes (CNS) equations within different relevant zones.
2. Generate data management connectivities that will allow viscous or inviscid formulation on an element-by-element basis.
3. Specify conditions to be applied at zonal interfaces.
4. Demonstrate the potential of using SCRAP technique for predicting resultant forces on a practical finned missile configuration with a gap between fin and body.
5. Document the results and conclusions of Phase I in a Final Technical Report.

All these Phase I objectives have been successfully completed for two spatial dimensions. Extension to three spatial dimensions will be carried out in Phase II. Although the development of an interactive block grid generator (IBGG) was not mentioned in the list of objectives, it was carried out as part of the effort to facilitate initial grid generation. Similarly the end product of Phase I was promised to be a feasibility study report that would establish the effectiveness of a numerical procedure that incorporates physics adaptation for distinct zones of a multiblock system. We have established this for subdomain adaptation as well. The methods used to achieve each part of the Phase I technical objectives are detailed in the following sections.

The Phase I effort successfully demonstrated the basic feasibility of simultaneous physics and resolution adaptation for multiblock grids. The technique provides significant savings in CPU time and storage over conventional structured grid algorithms employing a full NS solver over a fine grid of the same mesh size as the smallest grid size chosen by the spatial resolution adaptation. The SCRAP technique yields essentially the same results as the global calculation over a fine grid system. The results were also compared with previous computations and were found to be in good agreement. This successful demonstration of proof of concept in Phase I has

formed the basis for extension to 3-D in Phase II. The ultimate objective of the proposed effort is to deliver a complete and general purpose SCRAP software package that would be user friendly, modular, and would incorporate all the relevant physics pertaining to complex 3-D objects in flow fields encompassing subsonic to supersonic Mach numbers.

### 3. NUMERICAL DETAILS

This section describes the details of the methods utilized to achieve each part of the Phase I technical objectives. The section starts with the description of a general set of full Navier-Stokes equations which can be utilized as a physical model for the flow simulation. Subsection 3.2 describes the normalization of these equations. This is followed by the description of the inviscid and viscous terms in the numerical solver in terms of an unstructured finite volume cell-vertex scheme. The algorithm is shown to be decoupled into contributions from viscous and inviscid terms. Subsection 3.4 describes the implementation of boundary conditions. Subsection 3.5 presents a von Neumann stability analysis for a one-dimensional convection-diffusion model problem which suggests that different time steps can be used for the contributions from viscous and inviscid terms in the overall algorithm. Subsection 3.6 describes an interactive block grid generator that was developed as part of the Phase I effort. Subsection 3.7 describes the implementation of physics adaptation in the cells of interaction blocks.

A brief description of the new contributions and their significance is as follows:

- Development of an Interactive Block Grid Generator that allows grid splitting in user-defined blocks corresponding to specific schemes. The interior domain block-interfaces can be moved to generate "better" grids.
- Decoupling of the integration scheme into contributions from inviscid and viscous terms. This allows efficient implementation of the algorithm for viscous and inviscid blocks. Different time steps for these terms accelerate the convergence to steady state.
- Efficient implementation of physics adaptation for cells of interaction blocks. For these cells it is locally decided whether to use Euler or NS solver.
- Development of a general purpose code that efficiently implements physics and subdomain adaptation on multiblock grids.



### 3.1 Governing Equations

The complete Navier Stokes equations describe a general 3-D compressible viscous flow of a gas and are given by the following vector form

$$\frac{\partial \bar{U}}{\partial t} + \frac{\partial \bar{F}^i}{\partial x} + \frac{\partial \bar{G}^i}{\partial y} + \frac{\partial \bar{H}^i}{\partial z} = \frac{\partial \bar{F}^v}{\partial x} + \frac{\partial \bar{G}^v}{\partial y} + \frac{\partial \bar{H}^v}{\partial z} \quad (1)$$

where  $t$  represents the time and  $(x, y, z)$  the spatial coordinates as independent variables. The superscripts  $i$  and  $v$  signify inviscid and viscous terms, respectively. The state vector  $\bar{U}$ , inviscid flux vector  $\bar{F}^i$  and viscous flux vector  $\bar{F}^v$  are given by

$$\bar{U} = [\rho, \rho u, \rho v, \rho w, \epsilon]^T \quad (2)$$

$$\bar{F}^i = [\rho u, \rho u^2 + p, \rho uv, \rho uw, (\epsilon + p)u]^T \quad (3)$$

$$\bar{F}^v = [0, \tau_{xx}, \tau_{xy}, \tau_{xz}, u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - q_x]^T \quad (4)$$

where  $\rho$ ,  $p$ ,  $(u, v, w)$ , and  $\epsilon$  are the density, pressure, velocity components, and total energy per unit volume, respectively and  $T$  denotes the transpose operation. Furthermore  $\bar{\tau}$  and  $\bar{q}$  represent the stress tensor and heat flux, respectively. The representation of the flux vectors  $\bar{G}$  and  $\bar{H}$  is analogous to that of  $\bar{F}$  and is not shown here. Euler equations are obtained by setting the viscous flux vectors equal to zero.

Additional relations are required to complete the system of equations. An equation of state that connects pressure with total energy per unit volume is

$$\frac{\epsilon}{\rho} = \frac{v^2}{2} + c_p T - \frac{p}{\rho} = \frac{v^2}{2} + \frac{1}{\gamma - 1} \frac{p}{\rho} \quad (5)$$

where  $v$  is the magnitude of total velocity

$$v = \sqrt{u^2 + v^2 + w^2} \quad , \quad (6)$$

$c_p$  is the specific heat at constant pressure and  $T$  is the temperature of the gas. A calorically perfect gas is assumed for which the specific heat  $c_p$  is a constant.

The shear stress tensor  $\bar{\tau}$  is usually represented as a linear phenomenological law in terms of first and second viscosity coefficients ( $\mu$  and  $\lambda$ ). This in indicial notation is represented by

$$\tau_{ij} = \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \delta_{ij} \lambda \sum_{k=1}^3 \frac{\partial u_k}{\partial x_k} \quad (7)$$

where  $\delta_{ij}$  is the Kronecker delta

$$\delta_{ij} = 1 \quad \text{if } i=j; \quad 0 \text{ otherwise} \quad (8)$$

The first and second viscosity coefficients are usually connected by utilizing *Stokes hypothesis* which states that the sum of the diagonal elements of the shear stress tensor must equal zero. This hypothesis is obviously valid for a static fluid but is assumed to hold even for a fluid in motion and results in

$$\lambda = -\frac{2}{3}\mu \quad (9)$$

As an example, the viscous shear stresses for a 2-D flow are given by

$$\tau_{xx} = (\lambda + 2\mu) \frac{\partial u}{\partial x} + \lambda \left( \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) = \frac{2}{3}\mu \left( 2 \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right) \quad (10)$$

$$\tau_{yy} = (\lambda + 2\mu) \frac{\partial v}{\partial y} + \lambda \left( \frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} \right) = \frac{2}{3}\mu \left( 2 \frac{\partial v}{\partial y} - \frac{\partial u}{\partial x} \right) \quad (11)$$

$$\tau_{xy} = \mu \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \quad (12)$$

The dynamic viscosity coefficient  $\mu$  is a strong function of temperature  $T$  and is frequently modeled by the Sutherland law

$$\frac{\mu}{\mu_r} = \left( \frac{T}{T_r} \right)^{3/2} \left( \frac{T_r + S}{T + S} \right) \quad (13)$$

where the subscript r denotes a reference condition and S is a constant which is characteristic of a given gas. For air at moderate temperatures,  $S = 110.4$  K.

The heat flux  $\bar{q}$  is generally related to the thermal conductivity k by the Fourier law

$$\bar{q} = -k \nabla T = -k \left( \frac{\partial T}{\partial x} \bar{i} + \frac{\partial T}{\partial y} \bar{j} + \frac{\partial T}{\partial z} \bar{k} \right) \quad (14)$$

A constant Prandtl number is generally utilized as a relation between dynamic viscosity and thermal conductivity

$$Pr = \frac{C_p \mu}{k} \quad (15)$$

For air at normal conditions,  $Pr = 0.72$ .

### 3.2 Normalization of Governing Equations

Let the subscript r indicate some reference conditions and denote the non-dimensional quantities by asterisks, i.e., define

$$\begin{array}{lll} x = x^* L_r & y = y^* L_r & z = z^* L_r \\ t = t^* t_r & \rho = \rho^* \rho_r & \epsilon = \epsilon^* \epsilon_r \\ u = u^* u_r & v = v^* v_r & w = w^* w_r \\ p = p^* p_r & T = T^* T_r & \mu = \mu^* \mu_r \end{array} \quad (16)$$

In order to keep the form of the dimensional and normalized equations invariant, the continuity equation dictates

$$t_r = \frac{L_r}{u_r} \quad (17)$$

The inviscid part of the momentum flux terms yield

$$u_r^2 = \frac{p_r}{\rho_r} \quad (18)$$

whereas the inviscid part of energy flux term implies

$$\epsilon_r = p_r = \rho_r u_r^2 \quad . \quad (19)$$

The viscous flux terms of the momentum equations imply the following non-dimensional parameter of the fluid flows

$$Re = \frac{\mu_r}{L_r \rho_r u_r} \quad , \quad (20)$$

which is known as the Reynolds number. Thus the normalized form of Eqs. (1-4) remains invariant, except that the shear stress terms have to be multiplied by the factor  $Re^{-1}$  (i.e.,  $\mu$  is replaced by  $\mu^*/Re$ ). As an example, the normalized form of  $\tau_{xx}$  in Eq. (10) becomes

$$\tau_{xx}^* = \frac{2\mu^*}{3Re} \left( 2 \frac{\partial u^*}{\partial x^*} - \frac{\partial v^*}{\partial y^*} - \frac{\partial w^*}{\partial z^*} \right) \quad (21)$$

where  $\mu^*$  is given from the Sutherland law as

$$\mu^* = (T^*)^{3/2} \frac{1 + S/T_r}{T^* + S/T_r} \quad . \quad (22)$$

The non-dimensional heat flux term in the energy equation takes the form

$$\bar{q}^* = -\frac{\gamma}{\gamma-1} \frac{1}{Re} \left( \frac{\mu^*}{Pr} \right) \left( \frac{\partial T^*}{\partial x^*} \bar{i} + \frac{\partial T^*}{\partial y^*} \bar{j} + \frac{\partial T^*}{\partial z^*} \bar{k} \right) \quad . \quad (23)$$

Henceforth, the non-dimensional equations will be written without asterisks. Free stream conditions are used in this report as the reference basis. Hence, the normalized free stream values of density, pressure, and temperature will each be unity, whereas free stream velocity components will be

$$u_\infty = \sqrt{\gamma} M_\infty \cos \alpha \quad v_\infty = \sqrt{\gamma} M_\infty \sin \alpha \quad (24)$$

where  $\alpha$  is the angle of attack and  $M_\infty$  is the free stream Mach number. The normalized free stream total energy per unit volume from Eq. (5) is

$$\epsilon_\infty = \frac{\gamma M_\infty^2}{2} + \frac{1}{\gamma-1} \quad . \quad (25)$$

### 3.3 Integration Scheme

A new, conservative, finite volume scheme is developed for viscous (Navier-Stokes) equations as an extension of an earlier Euler scheme for unstructured quadrilateral grids. Since the flux balance is carried out on an element by element basis (i.e., no exterior properties are utilized for a given cell), the scheme is suitable for locally embedded unstructured grids. The viscous and inviscid changes for a given time step can be decoupled so that cells can be updated independently for these effects. This decoupling can allow substantial savings in computer resources by avoiding computations of viscous terms for the regions where they are negligible.

#### 3.3.1 Inviscid Terms

The integration scheme for inviscid terms has been previously introduced in Refs. (9-12). The integration scheme is based upon a second order accurate, Lax-Wendroff, finite volume scheme due to Ni.<sup>(13)</sup> Conservation variables  $U$  and fluxes  $F_i$ ,  $G_i$  are all stored at cell vertices and flux balance can be carried out for each cell independently for a given time step  $\Delta t$ .

Consider the flux balance for a cell  $C$  with general vertices  $i, j, k, l$ , as shown in Fig. 1, which results in the distribution formulas for the contribution of cell change to its corner nodes

$$\begin{aligned}\delta U_{iC}^i &= \frac{1}{4} \left[ \Delta U^i - \frac{\Delta t}{A} \Delta F - \frac{\Delta t}{A} \Delta G \right] \\ \delta U_{jC}^i &= \frac{1}{4} \left[ \Delta U^i + \frac{\Delta t}{A} \Delta F - \frac{\Delta t}{A} \Delta G \right] \\ \delta U_{kC}^i &= \frac{1}{4} \left[ \Delta U^i + \frac{\Delta t}{A} \Delta F + \frac{\Delta t}{A} \Delta G \right] \\ \delta U_{lC}^i &= \frac{1}{4} \left[ \Delta U^i - \frac{\Delta t}{A} \Delta F + \frac{\Delta t}{A} \Delta G \right] \quad .\end{aligned} \tag{26}$$

The superscript  $i$  signifies the inviscid terms, whereas subscripts like  $iC$  signify the contribution of cell  $C$  to node  $i$ . These distribution formulas allow for different time steps  $\Delta t$  and cell volumes  $A$  for elements adjoining a

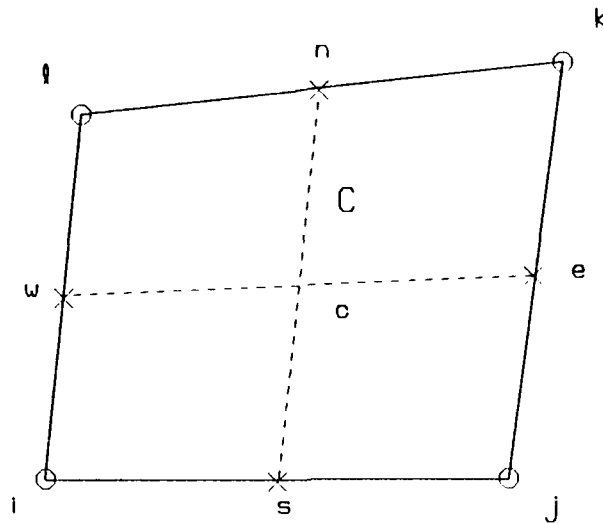


Figure 1. An Arbitrary Cell C with Nodes i, j, k, l.

common node. The cell change  $\Delta U^i$  in the above relations originates from a flux balance for cell C, i.e.,

$$\begin{aligned} \frac{A}{\Delta t} \Delta U^i &= F_w^i (y_l - y_i) - G_w^i (x_l - x_i) + F_n^i (y_k - y_l) - G_n^i (x_k - x_l) \\ &+ F_e^i (y_j - y_k) - G_e^i (x_j - x_k) + F_s^i (y_i - y_j) - G_s^i (x_i - x_j) \end{aligned} \quad (27)$$

where subscripts s, e, n, w indicate the south, east, north, west fluxes, respectively. Thus, for example

$$F_s^i = \frac{1}{2} (F_i^i + F_j^i) \quad (28)$$

The flux changes  $\Delta F$  and  $\Delta G$  are due to second order changes that provide stability to the algorithm and are given by

$$\begin{aligned} \Delta F &= (y_\eta \frac{\partial F^i}{\partial U} - x_\eta \frac{\partial G^i}{\partial U}) \Delta U^i \\ \Delta G &= (x_\xi \frac{\partial G^i}{\partial U} - y_\xi \frac{\partial F^i}{\partial U}) \Delta U^i \end{aligned} \quad (29)$$

The flux changes involve metrics  $x_\xi, y_\xi, x_\eta, y_\eta$ ; a typical example of a metric term for cell C is

$$y_\eta = \frac{y_n - y_s}{\Delta y_\eta} = \Delta y_{ns} = \frac{y_k + y_l - y_i - y_j}{2} \quad (30)$$

$$y_\eta = \frac{y_n - y_s}{\Delta y_{\eta s}} = \frac{y_k + y_l - y_i - y_j}{2} \quad (31)$$

The Jacobian matrix  $F_U$  is given by

$$F_U = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{\gamma-3}{2}u^2 + \frac{\gamma-1}{2}v^2 = F_{21} & (3-\gamma)u & (1-\gamma)v & \gamma-1 \\ -uv & v & u & 0 \\ u(F_{21} + u^2 - \frac{p+\epsilon}{\rho}) & \frac{p+\epsilon}{\rho} + (1-\gamma)u^2 & uv(1-\gamma) & u\gamma \end{bmatrix} \quad (32)$$

The Jacobian matrix  $G_U$  is similar to  $F_U$  and is not shown here.

### 3.3.2 Viscous Terms

A new scheme is presented here for treating the viscous terms which, like the inviscid scheme, utilizes only the properties interior to a given cell C. Only the first order contribution from the viscous terms is considered here to keep the computations within the cell C. Note that the only reason that second order terms were considered for the inviscid formulation was to impart stability to the numerical algorithm, since a first order Lax-Wendroff inviscid scheme is inherently unstable.

The flux balance around the boundary  $\partial C$  of cell C of Fig. 1 (See Eqs. (26-27)) yields

$$\frac{\Delta U}{\Delta t} = \frac{1}{A} \int_{\partial C} (F^i dy - G^i dx) - \frac{1}{A} \int_{\partial C} (F^v dy - G^v dx) \quad (33)$$

which can be rewritten as

$$\Delta U = \Delta U^i - \frac{\Delta t}{A} \int_{\partial C} (F^v dy - G^v dx) = \Delta U^i + \Delta U^v \quad (34)$$

where  $\Delta U^i$  has been evaluated in the previous subsection and  $\Delta U^v$  will be evaluated here. This equation clearly shows that the viscous and inviscid flux terms can be decoupled. It is apparent from Eq. (26) that the first order contribution to change at any of the corner nodes is one-fourth of the flux balance for the cell C. The same is also true for the viscous changes and has been rigorously proved in Ref. (11) for transformed coordinates. Thus the viscous terms can be handled in a simple manner once the viscous flux balance for the cell C is carried out. As an example of the evaluation of  $\Delta U^v$  consider the contribution from x-momentum equation. Thus,

$$\begin{aligned} \Delta U(2)^v = & \frac{\Delta t}{A} \{ \alpha_w (y_l - y_i) (2 \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y})_w - \beta_w (x_l - x_i) (\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})_w + \\ & \alpha_n (y_k - y_l) (2 \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y})_n - \beta_n (x_k - x_l) (\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})_n + \\ & \alpha_e (y_j - y_k) (2 \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y})_e - \beta_e (x_j - x_k) (\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})_e + \\ & \alpha_s (y_i - y_j) (2 \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y})_s - \beta_s (x_i - x_j) (\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})_s \} \quad (35) \end{aligned}$$

where (2) indicates the x-momentum equation contribution. The variable coefficients  $\alpha$  and  $\beta$  are given by

$$\alpha = -\frac{2}{3} \frac{\mu}{Re} \quad \beta = -\frac{\mu}{Re} \quad (36)$$

The determination of these coefficients at the appropriate faces is easy to formulate, e.g.,

$$\beta_s = -\frac{1}{2Re} (\mu_i + \mu_j) \quad (37)$$

where  $\mu$  is evaluated from Eq. (22). The determination of the gradient terms, like  $\frac{\partial u}{\partial x}|_w$ , introduces complications if one insists on a finite volume approach. One has to either employ a larger staggered stencil (thereby making the scheme unsuitable for unstructured grids)<sup>(14)</sup> or make approximations that the metrics and areas of the cells surrounding the given cell are nearly equal



to the corresponding quantities in cell C.<sup>(15)</sup> A new finite volume approach is presented here for unstructured grids in which such approximations are avoided by considering auxiliary staggered cells which lie completely within the cell C. These interior staggered cells are shown in Fig. 2. Thus for the determination of  $\frac{\partial u}{\partial x}|_e$  the cell jkns is used, however the line integration would only yield the gradient at the centroid  $ce$  of the staggered cell rather than at the edge node  $e$ . However, if values at the centroids  $ce$ ,  $cn$ ,  $cw$ , and  $cs$  of all the staggered cells can be determined, then one can extrapolate values at the edge nodes  $e$ ,  $n$ ,  $w$ , and  $s$ . As an example, consider the evaluation of  $\frac{\partial u}{\partial x}$  and  $\frac{\partial u}{\partial y}$  at the centroid  $ce$  by applying Green's theorem over the area  $ijew$ . The line integration around the auxiliary cell yields<sup>(14)</sup>

$$\left(\frac{\partial u}{\partial x}\right)_{cs} = \frac{1}{A/2} \left\{ \frac{u_w + u_i}{2} (y_i - y_w) + u_c (y_w - y_e) + \left(\frac{u_e + u_j}{2}\right) (y_e - y_j) + u_s (y_j - y_i) \right\} \quad (38)$$

$$\left(\frac{\partial u}{\partial y}\right)_{cs} = \frac{1}{A/2} \left\{ \frac{u_w + u_i}{2} (x_i - x_w) + u_c (x_w - x_e) + \left(\frac{u_e + u_j}{2}\right) (x_e - x_j) + u_s (x_j - x_i) \right\} . \quad (39)$$

Here the values  $u_s$ ,  $u_e$ , etc., are given by the averages

$$u_s = \frac{1}{2} (u_i + u_j) \quad (40)$$

$$u_c = \frac{1}{4} (u_i + u_j + u_k + u_l) . \quad (41)$$

Note that the area of the staggered cells has been approximated here as one-half the area of the cell C under consideration. This approximation will not hold for highly skewed cells. However, this is not a limitation of the overall approach since the actual areas of the staggered cells can be easily determined. These areas will have to be determined only once for the whole computation.

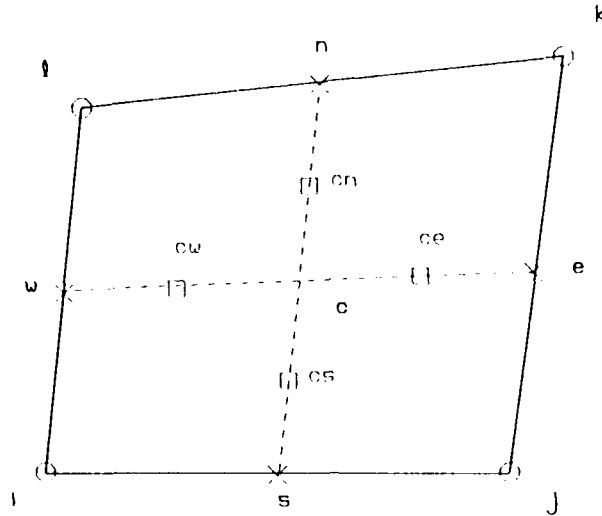


Figure 2. Staggered Cells for the Computation of Edge Gradients, Cell ijew is Used for  $\frac{\partial u}{\partial x}|_s$ , Cell isnl is Used for  $\frac{\partial u}{\partial x}|_w$ , Etc.

A biquadratic extrapolation can be used to determine values  $z_s$ ,  $z_e$ ,  $z_n$ , and  $z_w$  once some  $z$  values at the centroids of the auxiliary cells are determined. These extrapolations yield

$$\begin{aligned}
 z_s &= \frac{1}{4}(9z_{cs} + z_{cn} - 3z_{ce} - 3z_{cw}) \\
 z_c &= \frac{1}{4}(9z_{ce} + z_{cw} - 3z_{cs} - 3z_{cn}) \\
 z_n &= \frac{1}{4}(9z_{cn} + z_{cs} - 3z_{ce} - 3z_{cw}) \\
 z_w &= \frac{1}{4}(9z_{cw} + z_{ce} - 3z_{cs} - 3z_{cn}) \quad .
 \end{aligned} \tag{42}$$

The computations of viscous flux changes for cell C can then be determined by using an equation like Eq. (35). The viscous contribution to changes for nodes i,j,k,l is then

$$\delta U_{i,j,k,l} = \frac{1}{4} \Delta U^v \quad . \tag{43}$$

### 3.4 Boundary Conditions

The solution of the fluid dynamic equations is determined by the initial and boundary conditions. The initial condition for the cases in this report is typically taken as that corresponding to a uniform flow. A module that utilizes bilinear interpolation, based upon a previous solution (on a somewhat different grid), has also been developed to generate initial conditions for a given new grid system. Several different types of boundary conditions, such as those for inflow, outflow, rigid walls, symmetry walls, etc., are required for computing solutions. Two separate inflow/outflow boundary conditions are presented here.

#### 3.4.1 Free-Slip Rigid Walls

For inviscid flow, the appropriate physical condition on a solid surface is that there be no flow normal to the surface, or equivalently that the flow direction be tangential to the wall. The application of the free slip rigid wall boundary condition is explained in Fig. 3 for a node  $i$ , on a solid wall which makes a local angle  $\alpha$  with the x-axis.

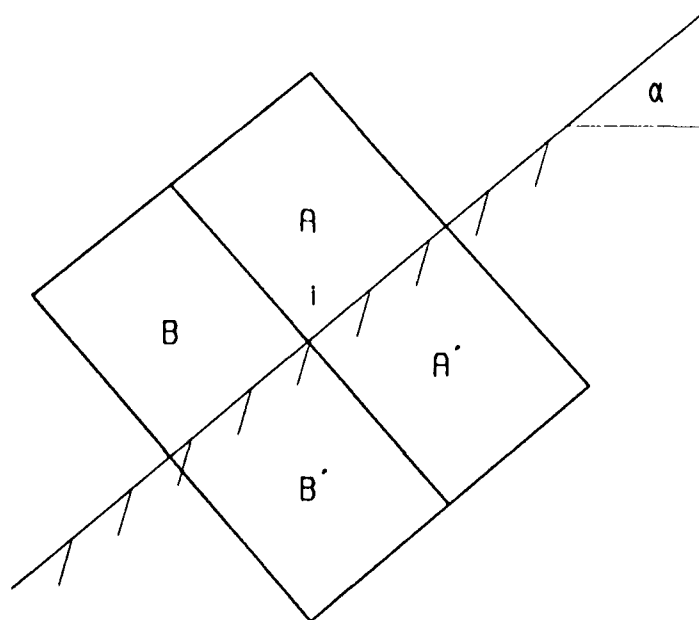


Figure 3. Physical Cells and their Images Adjacent to a Wall.

The application of the numerical scheme to cells A and B after integrating all the cells in the domain yields the following *initial change* at node i

$$\delta U_i = \delta U_{iA} + \delta U_{iB} \quad . \quad (44)$$

The *predicted change* at node i is taken to be that from cells A,B and their corresponding mirror images A',B', which contribute the same values, i.e.,

$$\delta U_i^P = 2\delta U_i \quad . \quad (45)$$

The superscript p indicates predicted change. If these values are not corrected, then the wall surface would be a line of symmetry for all variables, including of course, the normal component of the velocity. The local tangential component of the velocity is given by

$$V_t = u \cos \alpha + v \sin \alpha \quad , \quad (46)$$

and only this is used to reassign new velocity components along the coordinate directions, i.e.,

$$u = V_t \cos \alpha \quad , \quad v = V_t \sin \alpha \quad . \quad (47)$$

Thus, the corrected values for the changes are

$$\delta(\rho u)_i^C = (\rho V_t)^* \cos \alpha - (\rho u)_i^n \quad \text{Component 2} \quad (48)$$

$$\delta(\rho v)_i^C = (\rho V_t)^* \sin \alpha - (\rho v)_i^n \quad \text{Component 3} \quad (49)$$

$$\delta U_i^C = \delta U_i^P \quad \text{Other Components} \quad (50)$$

where

$$(\rho V_t)^* = [(\rho u)_i^n + \delta(\rho u)_i^P] \cos \alpha + [(\rho v)_i^n + \delta(\rho v)_i^P] \sin \alpha \quad (51)$$

here the superscript  $n$  indicates the values of the dependent variables at the previous time level. The superscript  $c$  indicates the corrected change which, when added to the old dependent variables, yields the values at the new time level  $n+1$ .

### 3.4.2 No-Slip Rigid Wall

The correct constraint at a viscous wall is the no-slip condition, which implies that the velocity components  $u, v$  vanish at a solid boundary. The temperature can either be prescribed (possibly isothermal wall), or its gradient can be set to specify a heat flux (possibly adiabatic wall). The predicted value of temperature can be obtained by solving the caloric equation of state and accounting for the *predicted change* in density and energy per unit volume, i.e.,

$$T^P = (\gamma - 1) \frac{\epsilon^{n+2\delta\epsilon}}{\rho^{n+2\delta\rho}} \quad (52)$$

Note that the corrected value of density is taken as

$$\rho^C = \rho^n + 2\delta\rho \quad (53)$$

The corrected value of temperature at the boundary node is

$$T^C = aT^P + (1-a)T_w \quad (54)$$

where  $a=1$  for an adiabatic wall and  $a=0$  for a given wall temperature  $T_w$ . The corrected value of energy per unit volume is then

$$\epsilon^C = \frac{T^C \rho^C}{\gamma - 1} \quad (55)$$

### 3.4.3 Inflow/Outflow Boundary Conditions

Two separate approaches are presented here. The first one applies the method of characteristics for Euler equations in a direction normal to the local grid surfaces. The second method applies the usual Riemann invariants

for isentropic flows at these boundaries. The former approach is somewhat complicated but should be valid for cases when strong shock waves intersect the inflow/outflow boundaries. Both the approaches neglect the viscous terms at the exterior flow boundaries. We first present the characteristic approach.

Let us transform the flow system such that the velocity  $u_\xi$  corresponds to that entering normal to the grid system and  $u_\eta$  corresponds to the tangential direction. This is shown in Fig 4. The following relations for the transformation of velocities hold

$$u_\xi = u \sin \alpha - v \cos \alpha \quad (56)$$

$$u_\eta = u \cos \alpha + v \sin \alpha \quad (57)$$

with the inverse relations

$$u = u_\xi \sin \alpha + u_\eta \cos \alpha \quad (58)$$

$$v = u_\eta \sin \alpha - u_\xi \cos \alpha \quad (59)$$

where  $\alpha$  is the angle which a grid boundary surface makes with x-axis.

The governing equations for the transformed coordinates are

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial \xi} + \frac{\partial G}{\partial \eta} = 0 \quad (60)$$

where only the inviscid flux vectors are considered. It is assumed that the boundaries are far enough away from the body that the viscous effects are small. Since the governing equations are *quasi-linear*, they can be written as

$$\frac{\partial U}{\partial t} + F_U \frac{\partial U}{\partial \xi} + G_U \frac{\partial U}{\partial \eta} = 0 \quad (61)$$

The Jacobian's  $F_U$  and  $G_U$  have been presented in Subsection 3.3. We will now neglect the variations along the tangential direction, i.e., set  $\partial U / \partial \eta = 0$ .

This is tantamount to assuming that the stream lines are locally straight and the velocity  $u_\eta$  in the tangential direction is a constant. If  $L$  denotes the matrix of eigenvectors of  $F_U$ , then

$$L \frac{\partial U}{\partial t} + (L F_U L^{-1}) L \frac{\partial U}{\partial \xi} = 0 \quad (62)$$

which can be written as

$$\frac{\partial Q}{\partial t} + \Lambda \frac{\partial Q}{\partial \xi} = 0 \quad (63)$$

where

$$Q = LU \quad (64)$$

is the characteristic variable vector in which  $L$  is assumed to be locally constant. Furthermore  $\Lambda = L F_U L^{-1}$  is a diagonal matrix whose elements are the eigenvalues of  $F_U$ .

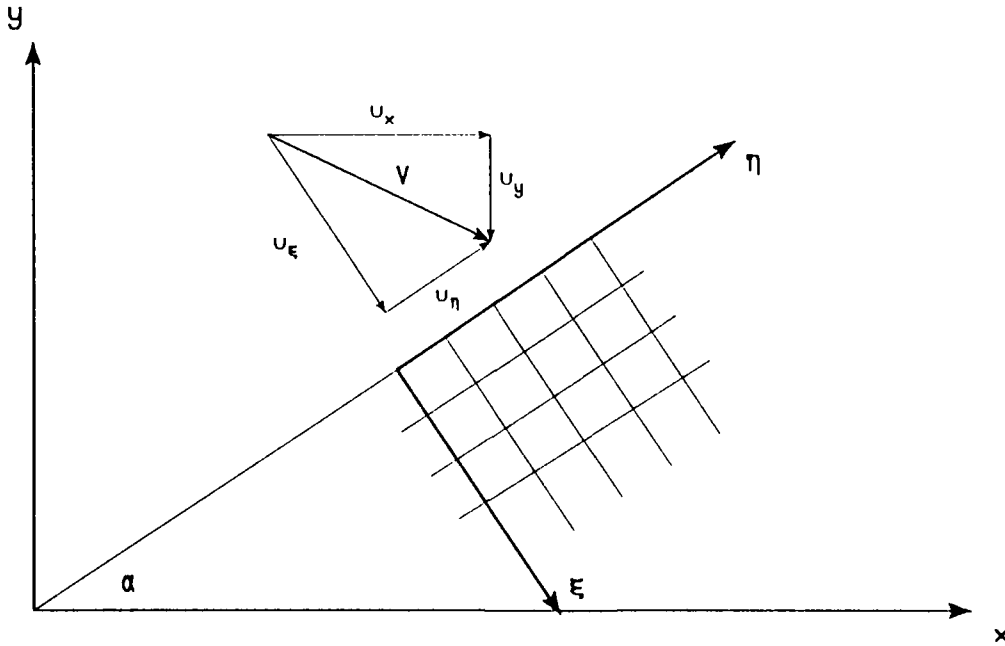


Figure 4. Coordinate Systems for Inflow/Outflow Boundaries.

The eigenvector matrix is given by

$$L = \begin{bmatrix} - & & & - \\ v^2 + \frac{au}{\gamma-1} & -(u + \frac{a}{\gamma-1}) & -v & 1 \\ \frac{v^2}{2} = \frac{au}{\gamma-1} & \frac{a}{\gamma-1} - u & -v & 1 \\ -v & 0 & 1 & 0 \\ \frac{u^2-v^2}{2} - \frac{a^2}{\gamma-1} & -u & 0 & 1 \\ - & & & - \end{bmatrix} \quad (65)$$

in which  $a$  is the speed of sound which is given by

$$a^2 = \frac{\gamma p}{\rho} \quad (66)$$

The corresponding diagonal matrix for the eigenvector matrix  $L$  is

$$\Lambda = \begin{bmatrix} - & & & - \\ u-a & 0 & 0 & 0 \\ 0 & u+a & 0 & 0 \\ 0 & 0 & u & 0 \\ 0 & 0 & 0 & u \\ - & & & - \end{bmatrix} \quad (67)$$

Equation (63) represents a decoupled system of equations with the differential change in characteristic variables given by

$$dQ = LdU \quad (68)$$

or



$$\begin{array}{cccc|c|c|c}
- & & & & - & - & - \\
v^2 + \frac{au_\xi}{\gamma-1} & -(u_\xi + \frac{a}{\gamma-1}) & -u_\eta & 1 & \rho & & u_\xi - a \\
\frac{v^2}{2} - \frac{au_\xi}{\gamma-1} & \frac{a}{\gamma-1} - u_\xi & -u_\eta & 1 & \rho u_\xi & & u_\xi + a \\
-u_\eta & 0 & 1 & 0 & d\rho u_\eta & \text{with} & u_\xi \\
\frac{u_\xi^2 - u_\eta^2}{2} - \frac{a^2}{\gamma-1} & -u_\xi & 0 & 1 & \epsilon & & u_\xi \\
- & & & & - & - & -
\end{array} \quad (69)$$

The third characteristic variable (third row) is the easiest and it yields  $\rho du_\eta = 0$  or that  $u_\eta$  is a constant along the characteristic direction  $\frac{d\xi}{dt} = u_\xi$ . Other characteristic variables yield (after some simplification)

$$dq_1 = \frac{dp}{\gamma-1} - \frac{\rho a}{\gamma-1} du_\xi = 0 \quad \text{along } u_\xi - a \quad (70)$$

$$dq_2 = \frac{a\rho}{\gamma-1} du_\xi + \frac{dp}{\gamma-1} = 0 \quad \text{along } u_\xi + a \quad (71)$$

$$dq_3 = \rho du_\eta = 0 \quad \text{along } u_\xi \quad (72)$$

$$dq_4 = \frac{dp}{\gamma-1} - \frac{a^2}{\gamma-1} d\rho = 0 \quad \text{along } u_\xi \quad (73)$$

Here  $q_j$  ( $j=1,2,3,4$ ) are regarded to be the elements of the characteristic vector variable  $Q$ . Assuming the coefficients of the above equations to be locally frozen (values obtained prior to updating), these equations can be easily integrated. The frozen values are shown here to be subscripted by letter  $c$  and the characteristic constants are

$$q_1 = p - \rho_c a_c u_\xi \quad (74)$$

$$q_2 = p + \rho_c a_c u_\xi \quad (75)$$

$$q_3 = u_\eta \quad (76)$$

$$q_4 = p - a_c^2 \rho \quad (77)$$

Since the process of updating ( $U^{n+1} = U^n + \delta U$ ) involves the contributions to nodes from the interior domain, the values based upon  $U^{n+1}$  should be used for characteristics inferred from the interior computational domain. Similarly, the characteristics based upon external conditions should either utilize free stream values or values based upon  $U^n$ . Wherever these characteristic variables are inferred from, the primitive variables can be computed from the following relations

$$p = \frac{q_1 + q_2}{2} \quad (78)$$

$$u_\eta = q_3 \quad (79)$$

$$u_\xi = \frac{q_2 - q_1}{2\rho_c a_c} \quad (80)$$

$$\rho = \frac{q_1 + q_2 - 2q_4}{2a_c^2} \quad (81)$$

For each type of characteristic boundary condition,  $\rho_c$  and  $a_c$  are determined from  $U^n$ . The specific boundary condition prescription depends upon whether the flow field is subsonic or supersonic.

For supersonic inflow all the characteristic information comes from the exterior, and it is implemented by setting the changes  $\delta U_i$  at such nodes equal to zero. For supersonic outflow all the characteristics are inferred from the interior, and the changes predicted by the numerical scheme itself can be used. Thus, no special treatment is needed for supersonic exit boundaries. For subsonic inflow,  $q_1$  is inferred from the updated  $U^{n+1}$  values and  $q_2, q_3, q_4$  are based upon free stream values. For subsonic exit  $q_1$  is computed from

free stream values (or other specified exit conditions), whereas  $q_2, q_3, q_4$  are computed from updated  $U^{n+1}$  values.

We now describe the alternate inflow/outflow boundary conditions based upon Riemann invariants. For the subsonic exit the outflow pressure  $p_1$  is specified. For supersonic flow with some imbedded subsonic regions this pressure is inferred from the supersonic portion of the flow (at the place where the Mach number just becomes supersonic). Let  $\rho_2, u_2, v_2, p_2$  be the flow properties at some node interior to the computational domain. The density at the outflow node can be determined from the constant entropy condition

$$\rho_1 = \rho_2 \left( \frac{p_1}{p_2} \right)^{1/\gamma} \quad (82)$$

The velocity  $u_1$  can be determined from the  $R_+$  invariant which is a constant along the characteristic  $\frac{dx}{dt} = u+a$ , i.e.,

$$u_1 = R_+ - \frac{2}{\gamma-1} \sqrt{\gamma p_1 / \rho_1} \quad (83)$$

where

$$R_+ = u_2 + \frac{2}{\gamma-1} \sqrt{\gamma p_2 / \rho_2} \quad (84)$$

The velocity  $v_1$  is determined from the tangency condition

$$v_1 = v_2 \left( \frac{u_1}{u_2} \right) \quad (85)$$

Once  $\rho_1, u_1, v_1, p_1$  are known, the energy  $e_1$  can be determined from the caloric equation of state (5).

For the subsonic inflow boundary, total pressure, total temperature and flow angle  $\frac{v}{u}$  can be specified while the Riemann invariant  $R_-$  can be extrapolated from the interior of the flow. Note that

$$R_- = u - \frac{2}{\gamma-1} \sqrt{\gamma p / \rho} \quad (86)$$

is a constant along the characteristic  $\frac{dx}{dt} = u-a$ .

### 3.5 Stability Analysis

An important step in the development of new algorithms is the determination of time step restrictions through a stability analysis. This section explores the choice of a time step for the explicit schemes presented in Subsection 3.3 through a one-dimensional convection-diffusion model equation by utilizing von Neumann stability analysis. Since the viscous and inviscid term contributions have been separated out in Subsection 3.3, different time steps can, in principle, be applied to these terms. The generalization to two-spatial dimensions is also carried out in this section.

#### 3.5.1 One-dimensional Convection-diffusion Model

The one-dimensional x-momentum equation, as presented in Subsection 3.3, can be written in the following non-dimensional form

$$\frac{\partial}{\partial t}(\rho u) + \frac{\partial}{\partial x}(\rho u^2 + p) = \frac{\partial}{\partial x}(\tau_{xx}) = \frac{\partial}{\partial x} \left\{ \frac{4\mu}{3Re} \frac{\partial u}{\partial x} \right\} \quad (87)$$

Considering the viscosity  $\mu$  to be a constant and subtracting off the continuity equation, this simplifies to

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + \frac{1}{\rho} \frac{\partial p}{\partial x} = \frac{4\mu}{3\rho Re} \frac{\partial^2 u}{\partial x^2} \quad (88)$$

The model equation is considered to be similar to this and to have a constant wave speed  $c$ , i.e.,

$$\frac{\partial U}{\partial t} + c \frac{\partial U}{\partial x} = \nu \frac{\partial^2 U}{\partial x^2} \quad (89)$$

where one can regard the constant  $\nu$  to be proportional to the kinematic viscosity

$$\nu = \frac{4\mu}{3\rho Re} \quad (90)$$

Discretizing Eq. (89) according to a Lax-Wendroff<sup>(16)</sup> scheme yields

$$\begin{aligned}
u_i^{n+1} - u_i^n &= \frac{\Gamma}{2} \{ (u_{i-1} - u_{i+1}) + \frac{\Gamma^2}{2} (u_{i-1} - 2u_i + u_{i+1}) \} \\
&\quad + \Sigma (u_{i-1} - 2u_i + u_{i+1})
\end{aligned} \tag{91}$$

where

$$\Gamma = \frac{c\Delta t}{\Delta x^2} \tag{92}$$

is the CFL (Courant-Friedrichs-Lewy) number and

$$\Sigma = \frac{1}{R_c} = \frac{\nu\Delta t}{\Delta x^2} \tag{93}$$

is the reciprocal of the cell Reynolds number, which shall be referred to here as the *cell diffusion number*. Applying the standard von Neumann stability analysis to Eq. (91) yields the following constraint

$$\Gamma^2 + 2\Sigma \leq 1 \tag{94}$$

This constraint is a combination of the CFL restriction

$$\Gamma \leq 1 \tag{95}$$

which holds for inviscid flows, and a diffusion stability limitation

$$\Sigma \leq \frac{1}{2} \tag{96}$$

which states that the cell-Reynolds number has to be greater than 2 when the convection term is negligible. On account of Eq. (95), the combined constraint for the convection-diffusion problem can be revised to yield a more restrictive form

$$\Gamma + 2\Sigma \leq 1 \tag{97}$$

which yields the following expression for time steps<sup>(17)</sup>

$$\Delta t \leq \frac{\Gamma \Delta x}{c + v / \Delta x \Sigma} \quad (98)$$

However, since the viscous and inviscid parts can be integrated out separately for Ni scheme, there is no need to implement the above restrictive form on the two contributions. The inviscid time step follows from the CFL restriction (Eq. (95))

$$\Delta t_i \leq \frac{\Delta x}{u+a} \quad (99)$$

where  $u$  is the flow velocity and  $a$  is the speed of sound, whereas the subscript  $i$  denotes inviscid time step. Note that the constant wave speed  $c$  has been replaced by the maximum eigenvalue,  $u+a$ , of the 1-D system. The viscous time step follows from the diffusion restriction

$$\Delta t_v \leq \frac{1}{2} \frac{\Delta x^2}{v} = \frac{3\rho Re \Delta x^2 \Sigma}{4\mu} \quad (100)$$

where the subscript  $v$  indicates the viscous time step. Hence the ratio of the two time steps is given by

$$\frac{\Delta t_v}{\Delta t_i} = \frac{3\rho Re \Delta x (u+a) \Sigma}{4\mu \Gamma} \quad (101)$$

Utilizing the normalized free stream values for these quantities, this ratio of the two stable time steps can be shown to be

$$\frac{\Delta t_v}{\Delta t_i} \simeq \frac{3}{8} Re \Delta x \sqrt{\gamma} (1+M_\infty) \quad (102)$$

where  $\gamma$  is the ratio of specific heats and  $M_\infty$  is the free-stream Mach number. Thus for a large Reynolds number flow, the viscous time step can be orders of magnitude larger than the inviscid time step. For such a flow, if one utilizes a conventional coupled scheme in which the time step  $\Delta t$  is the minimum of  $\Delta t_i$  and  $\Delta t_v$  (or a hyperbolic mean as suggested by Eq. (98)), then it will take substantial computer resources to converge on the viscous regions. In the spirit of local time stepping, the viscous and inviscid contributions can be updated at their respective time restrictions for the

uncoupled algorithm developed in Subsection 3.3. This will allow the algorithm to converge significantly faster compared to conventional uncoupled schemes.

### 3.5.2 Time Steps in 2-D

The stability limit for the linearized Euler equations from the analysis of a 2-D wave equation yields<sup>(11)</sup>

$$\frac{\Delta t_i}{\Gamma A} = \min \left\{ \frac{1}{|u\Delta y_\eta - v\Delta x_\eta| + aD_\eta}, \frac{1}{|u\Delta y_\xi - v\Delta x_\xi| + aD_\xi} \right\} \quad (103)$$

where A is the area of a cell under consideration and

$$D^2 = \Delta x^2 + \Delta y^2 \quad (104)$$

The quantities  $\Delta y_\eta = \Delta y_{ns}$ ,  $\Delta y_\xi = \Delta y_{ew}$ , etc., are the metrics as presented in Subsection 3.3.

The generalization for the viscous time steps is also straight-forward, i.e.,

$$\Delta t_v = \frac{3\rho Re}{4\mu} A \Sigma \min \left\{ \frac{D_\xi}{D_\eta}, \frac{D_\eta}{D_\xi} \right\} \quad (105)$$

### 3.5.3 Actual Implementation of Time Steps

The system of equations becomes numerically stiff for very high Reynolds number. It is clear from Eq. (102) that for such a case the viscous and inviscid time steps become highly disparate. Since these time step restrictions have been derived from a linearized analysis, it is no surprise that the computations become unstable for non-linear systems of equations for high Reynolds number flows. The stiffness problem is addressed in this study by computing a maximum time step for each cell

$$\Delta t_m = F \min (\Delta t_i, \Delta t_v) \quad (106)$$

where  $F$  is a large factor that keeps  $\Delta t_i$  and  $\Delta t_v$  to within the same order of magnitude; a maximum value of  $F=10$  will accomplish this. The inviscid and viscous time steps are then adjusted as

$$\Delta t_i = \max(\Delta t_i, \Delta t_m) \quad (107)$$

$$\Delta t_v = \max(\Delta t_v, \Delta t_m) \quad (108)$$

Note that the above ad-hoc recipe for time steps still uses different values for viscous and inviscid terms, but limits their disparity to within an order of magnitude variation.

### 3.6 Interactive Block Grid Generator

The generation of initial grids for complex geometries can be a difficult task. The grid generation for even simple flow fields with multiple embedded solid objects can be troublesome. The issue of the complexity of the grid generation can be partially simplified by considering a multiblock gridding technique. Subdomain adaptation or automatic division of cells in the regions of enhanced gradients of selected flow field variables, can be used to further simplify this process.

For an initial grid generation an interactive block grid generator (IBGG) has been developed as part of the Phase I effort. The IBGG subdivides the flow domain into blocks which are simply connected regions. Each of these blocks is associated with a given numerical scheme. These blocks are classified as Euler, NS, TLNS, or Interaction, depending upon the scheme being utilized. In the interaction blocks a decision is locally made whether to use Euler or NS equations.

In the IBGG approach each block of the overall configuration initially has an independent grid structure. The topology of one block has no bearing on the rest of the blocks, except that there must be a node-to-node matching across the interfaces of contiguous blocks. The user specifies the nodes of the boundaries of the blocks and a check is made for node-to-node matching at inter-block interfaces. Although overlapping interzonal interfaces provide maximum geometrical flexibility for complex configurations, such overlapping



meshes were not considered for this research. Overlapping algorithms are non-conservative in the sense that interpolation schemes are needed to accomplish intergrid property transfer. Furthermore, the logic of coupling an overlapping grid scheme with a physics and subdomain adaptation would have become unduly complicated.

Figure 5 shows the first step in the interactive block grid generation. The geometry corresponds to an 8% circular arc cascade with a flap placed near the trailing edge of the arc. The user has selected 14 blocks. The circles correspond to the user selected node points of the block interfaces. The user has clustered the node points near the solid surfaces. Only part of the overall block system is shown. As shown in Fig. 6, the user also specifies the numerical schemes to be used on individual blocks. Here E stands for Euler scheme, V for a viscous scheme, and I for an interaction block.

For the developed IBGG, the geometry of each block face can be specified in terms of cubic polynomials. Utilizing the interblock boundary point information, the interior grid of each of the blocks is generated via an algebraic grid generator. This step also determines the connectivity arrays for each cell, node and boundary point in each block. Note that additional nodes will exist at the time of assembly of the overall grid when the points on the contiguous boundaries coincide. These multiply defined nodes are

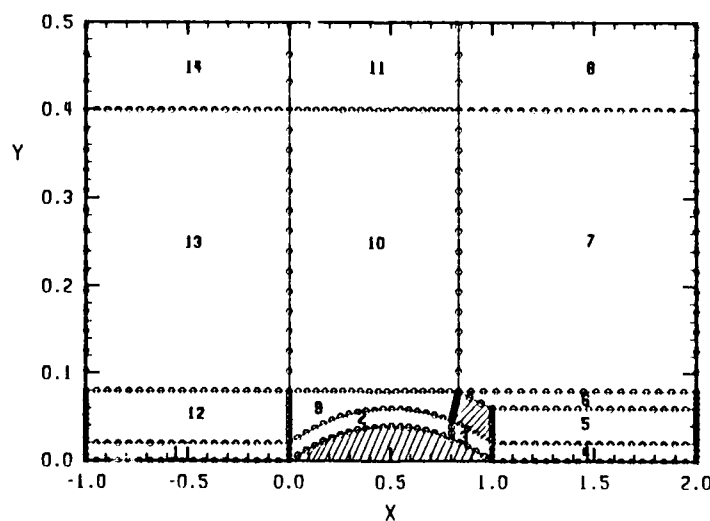


Figure 5. User Selected Node Points for 14 Blocks in a Computational Domain of an 8% Circular Arc Cascade with a Flap.

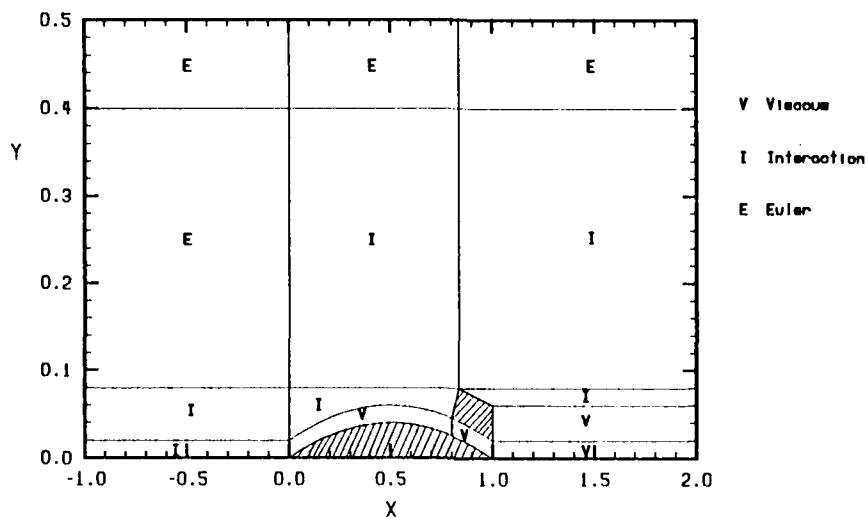


Figure 6. User Selected Numerical Schemes for the 14 Blocks in the Computational Domain of an 8% Circular Arc Cascade with a Flap.

marked for deletion, and the connectivity arrays of the interface nodes are examined and adjusted for consistency in the data structure. Note that the total number of cells in the overall grid remains invariant as the blocks are assembled. Figure 7 shows the crude base mesh generated by IBGG after the final assembly of inter-block interfaces. At this point, the data base forgets about the blocks and all the alignments are in terms of individual cells.

After the generation of the crude assembly all the cells with the same numerical algorithm are stored together in the computer memory. These cells need not be physically contiguous. At this stage computations can be carried out on the crude grid since all the connectivity arrays are defined and debugged. Alternatively, the crude assembly can be filtered through an elliptic grid generator to move the nodes appropriately. This filtering process does not change the connectivity arrays but rather the spatial coordinates of the nodes. An elliptic grid generator has been developed that utilizes the pointer system of the unstructured grid connectivity arrays. In its present form, it fixes the locations of the nodes on the physical boundaries, but allows the interior nodes (including those which were originally at block interfaces) to be displaced. Reference (18) presents an

elliptic grid procedure for structured multiblock grids where special treatment is needed at the block interfaces. No such special treatment is needed for the overall unstructured data base utilized here.

Figure 8 shows the base grid after the IBGG has processed the previous crude assembly through an elliptic grid generator which had no forcing functions. For this figure, the nodes at the physical boundaries were held fixed and the absence of forcing functions caused local distortions of the grid lines near these boundaries. Reference (17) presents a procedure for the determination of the interior grid forcing function values based upon their values at the physical boundaries. Further evaluation of forcing functions will be carried out in Phase II.

After the final grid assembly, regions of cells (irrespective of blocks) can be manually subdivided to incorporate pre-embedding. Thus cells near solid boundaries can be manually divided to capture boundary layers. Such pre-embedding can result in enhanced spatial resolution without introducing grid lines in the regions where they are not needed. Each cell can also be spatially divided by the algorithm to incorporate subdomain adaptation. Physics adaptation may be carried out in all the cells which previously belonged to interaction blocks.

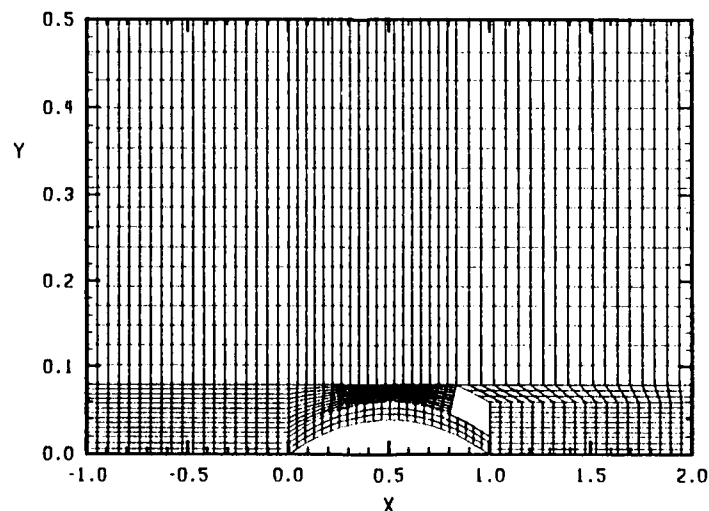


Figure 7. Initial Interior Grid Assembly for the Computational Domain of an 8% Circular Arc Cascade with a Flap.

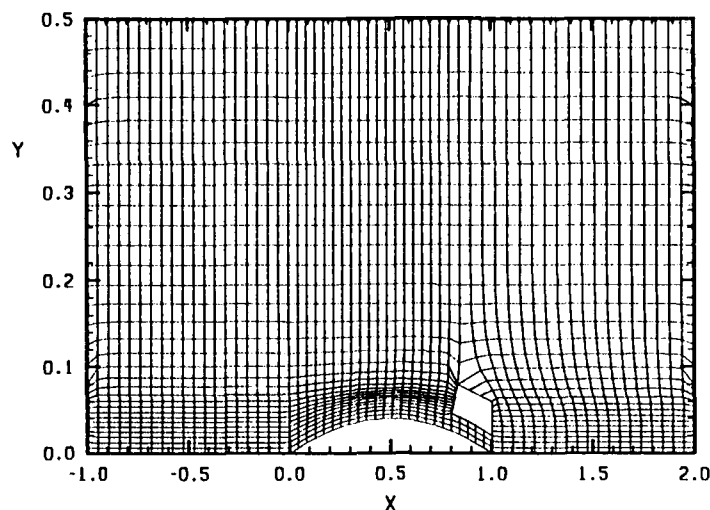


Figure 8. Grid Assembly Produced by an Elliptic Grid Generator for the Computational Domain of an 8% Circular Arc Cascade with a Flap.

For the IBGG, various stages of the grid assembly or its parts can be graphically viewed and debugged interactively. The user could request the program for specific changes. For example, the user may move nodes in certain regions, subdivide or fuse meshes, examine or query specific connectivities of various elements (cells, nodes, boundary points), etc.

### 3.7 Physics Adaptation

The multiblock SCRAP technique simplifies physical complexity by associating a numerical scheme to each of the blocks. Thus, one can have viscous blocks where NS equations are solved and inviscid blocks where one carries out the solution to Euler equations. Such a scheme-splitting technique can efficiently model complex physics if the domains of viscous and inviscid regions are known a priori. The SCRAP technique addresses this issue by considering interaction blocks where a decision is locally made whether to use Euler or NS equations. This dynamic physics switching is termed here as *physics (or equation) adaptation*. The interaction blocks will be important for the cases where the viscous/inviscid interaction region develops as part of the numerical solution.

There are numerous advantages for utilizing viscous/inviscid interaction blocks in addition to separate inviscid and viscous blocks. For certain applications, the user may know a priori exactly where the inviscid and viscous parts are located. For example, in a missile configuration, the "outer" region is always inviscid, and the flow near the surfaces or boundary layers is always viscous. For such distinct regions, computer resources should not be wasted in deciding which numerical scheme to use. Most flow fields also possess regions where it is not clear whether to use a viscous or inviscid formulation. Such regions should be selected as interaction blocks. Physics adaptation will be beneficial for these intermediate blocks that form the buffer between viscous and inviscid blocks. If such interaction blocks are not considered, then one has to generally treat such regions as being fully viscous, and this may be a waste of resources. In addition, if viscous and inviscid blocks are directly contiguous, then special treatment may be needed at block interfaces to maintain conservation. It can be shown that such treatment is not needed if interaction blocks are regarded as buffer zones between viscous and inviscid blocks.

As shown in Subsection 3.3, the overall algorithm is decoupled into contributions from viscous and inviscid terms. This means that Euler equations are solved everywhere, including the viscous region. This is followed by the determination of viscous cells in the interaction blocks. These cells are lumped together with cells in the viscous regions. Additional viscous contributions (corresponding to NS or TLNS equations) are then computed for these viscous cells and added to the inviscid contributions. The decoupled nature of the algorithm allows different time steps to be taken for viscous and inviscid contributions. This is in line with the concept of local time stepping for carrying out solutions to steady state problems. For these steady state applications, the determination of viscous cells in the interaction blocks can be carried out once every few hundred iterations. This means that the CPU time associated with the process of selection of numerical schemes will be small compared to the CPU time associated with the integration of cells. For unsteady applications where the viscous/inviscid interaction regions may grow significantly as part of the numerical solution, physics adaptation may have to be carried out more frequently.

Physics adaptation is incorporated by finding the maximum shear stress in the cells of viscous and interaction blocks. Those cells of the interaction block for which the shear stress is more than 1% of the maximum shear stress are regarded as viscous cells. Since the shear stresses only change by a very small amount near the interfaces between the contiguous viscous and inviscid cells, a special treatment is not needed at these interfaces. The viscous and inviscid contributions (respectively,  $\delta U_j^V$  and  $\delta U_j^I$  for all nodes  $j$ ) are separated out for each node. For those cells which are only regarded as inviscid, the viscous contributions to its corresponding nodes is regarded zero. Obviously at the interface between viscous and inviscid cells, some nodes will have viscous contributions from viscous cells and the same nodes will have no such contribution from inviscid cells. After these contributions are summed, the process of updating is carried out by adding the overall contributions to the previous values of the dependent variables.

It is appropriate to point out that the placement of the zonal interface between inviscid and viscous solvers is very important. If the interface is located in a region that is characterized by strong flow gradients that are not modeled by Euler equations, then there would appear to be problems with the overall solution. This is not really a shortcoming of the zonal gridding technique but rather the fact that important physics is being neglected by using the inappropriate mathematical models. Hence care must be exercised in examining the overall geometrical configuration for the initial generation of subdomain grids and the allocation of numerical schemes to these blocks. It would seem logical to examine the relative magnitude of the stress tensor in the inviscid blocks, after a specified number of iterations, to confirm the validity of the inviscid part. If the stress values are found to be substantial, then the overall algorithm should be able to reassign the numerical schemes in corresponding subdomains.

As part of a feedback from ARO, a local basis for the selection of viscous cells in the interaction blocks has been suggested. Thus, one may compare the magnitude of viscous terms with convection (or pressure) terms locally to select the viscous cells. Although the global approach has worked fine for the selected model problem, its disadvantage is that two passes are needed to accomplish the objective. In the first pass, maximum viscous stress is determined (from viscous and interaction blocks) and the second pass tags

the cells in the interaction region. For the local approach only a single pass over the interaction blocks cells will be needed to tag the cells appropriately. Since the suggested approach was discussed towards the end of the contract period, the Principal Investigator (PI) was unable to evaluate it. The evaluation and utilization of a local approach to determine viscous cells will be carried out in Phase II.

### 3.8 Subdomain Adaptation

The SCRAP technique also implements *subdomain adaptation* in which the algorithm automatically implements additional spatial resolution by locally dividing cells in the regions of enhanced gradients of selected flow field variables. The details of this process have appeared in Refs. (9)-(12). This multi-variable approach defines a single scalar criterion variable, based upon the first differences of multiple components, that removes the effect of inter-correlations between individual components. This measure allows an unbiased spread of data for the cases when the variabilities in different components are different and when some or all these components are correlated. Since the single scalar variation is a transformation to the *standardized form*, spatial domains characterized by different kinds of variabilities can be adapted by using the same procedure. Thus the same approach can be used to adapt in the viscous and inviscid regions.

The cells for which the standardized variation exceeds a threshold limit are locally subdivided. The cells for which the variations remain below another smaller threshold value are allowed to merge. First differences of density were used to adapt in the inviscid regions near shock waves, whereas the first differences of x-velocity were found to be suitable to capture the boundary layer.

#### 4. PHASE I NUMERICAL RESULTS

This section describes the numerical results of Phase I. The potential of the SCRAP technique has been demonstrated for a supersonic flow over an 8% circular arc cascade. This model problem is a benchmark case, and it is used to test the current approach. Results have been obtained by utilizing various combinations of different schemes on multiple zones (plus varying time steps for viscous and inviscid schemes), physics adaptation and subdomain adaptation. These results were compared with those utilizing full NS equations on a single fine grid and yielded essentially the same answers for the same boundary conditions. It was also found that utilization of the characteristic approach versus the Riemann invariant approach for the subsonic outflow boundary yielded somewhat different results. The results for the subsonic outflow Riemann invariant approach, utilizing subdomain and physics adaptation, compared very well with previous computations of other researchers.

The variation of the flow properties across the zonal interfaces was found to remain independent of the interface itself. The fluid properties also remain oblivious to those interfaces across which the grid changes geometrically or different schemes are used. The technique provides significant savings in CPU time and storage over conventional structured grid algorithms employing a full NS solver. An increase of a factor of about 50 in computational speed was achieved when compared to a conventional full NS solution over a fine grid of the same mesh size as the smallest grid size chosen by the spatial resolution adaptation. Further savings can be produced by considering additional levels of subdomain adaptation to capture small scale features. This demonstration clearly establishes that simultaneous physics and resolution adaptation works for the sample problem.

##### 4.1 Model Problem

The potential of the SCRAP technique can be demonstrated by considering a model problem in two spatial dimensions. Figure 9 shows the geometry and initial fine grid for such a model problem. The distances are normalized by



the chord length for this 8% circular arc cascade. The free stream Mach number is 1.4, and the Reynolds number, based upon the chord length, is 23,000. There are 51 points along the vertical and 173 points along the horizontal (8823 nodes). This model problem was selected because the solution for both Euler and NS equations is well known. A single fine grid was initially chosen to test the algorithm and so that comparisons of CPU time can be made with other adaptive solutions. Thus there are initially no blocks for the solutions of Euler and full NS equations in this single grid.

Supersonic inlet boundary condition is applied at  $x=-1$ , whereas characteristic outflow condition is specified at  $x=2.0$ . For the most part, the exit condition is supersonic with a subsonic region near  $y=0$ . The top wall is considered to be inviscid solid slip wall with no vertical velocity. The bottom wall is a slip wall for Euler equations, and a no-slip wall for NS equations. Except where noted otherwise, the boundary condition for the subsonic outflow is based upon the characteristic analysis for both viscous and inviscid solutions.

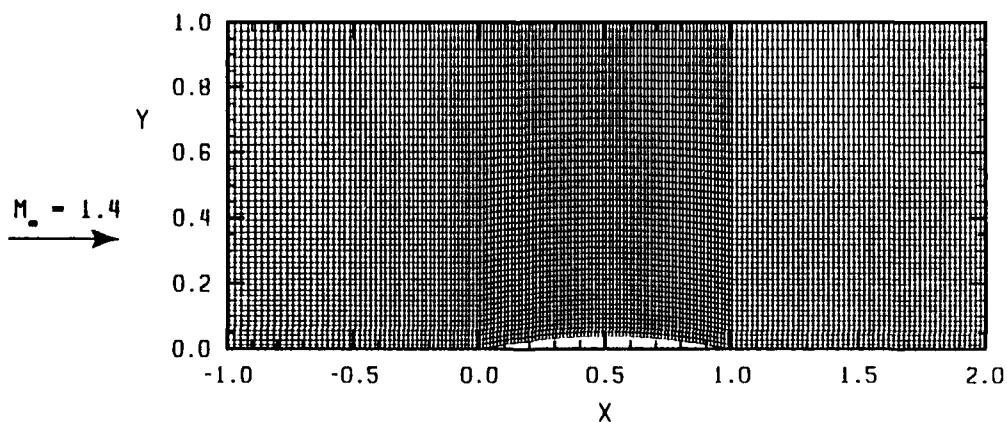


Figure 9. Geometry and Initial Grid (51X173) For Supersonic Flow Over an 8% Circular Arc Cascade.

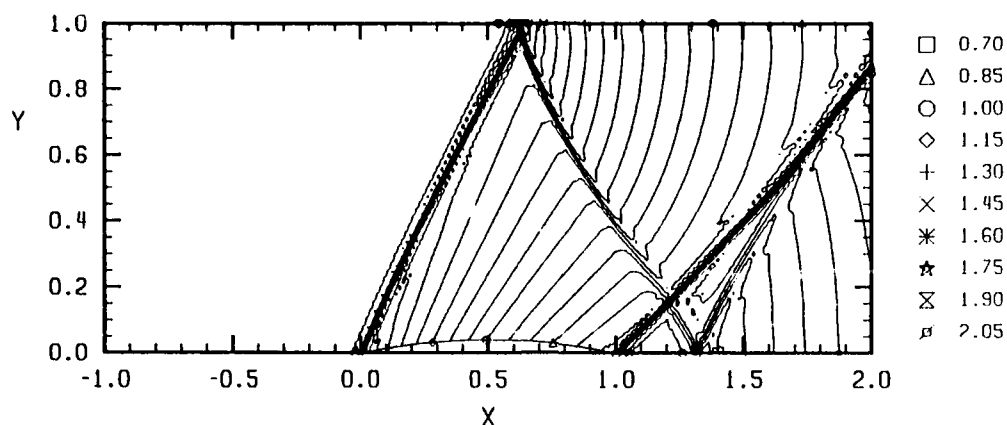


Figure 10. Density Contours for the Euler Solution for 8% Circular Arc Cascade on a 51x173 Mesh System. The Increment for Density Contours is 0.05.

#### 4.2 Euler Solution

Figure 10 presents the Euler solution for the previous geometry and grid. It is noted that the scheme has good shock capturing characteristics. This solution agrees very well with those of other investigators.<sup>(19,20)</sup> The incident shock is reflected back as a regular reflected shock from the top boundary. This reflected shock intersects with the trailing edge shock and also gets reflected at the lower surface. The supersonic outflow boundary condition is clearly appropriate since the the trailing edge shock is absorbed by the exit boundary rather than being reflected. The total CPU time for this inviscid solution on SSI's Data General computer was 72 minutes when the RMS residuals had subsided to below  $10^{-5}$ .

#### 4.3 Full Navier-Stokes Solution

Figure 11 represents the full NS solution for  $Re=23000$  over the same grid of 51x173 mesh points. The NS solution is distinctly different from the Euler solution. The boundary layer changes the flow field, and this means that the incoming flow now encounters an effectively larger obstruction. This has the effect of changing the flow field in the inviscid part. For example near  $y=1$ , instead of a regular reflection, a Mach disc formation is observed. The

reflected shock interacts with the boundary layer and results in shock wave/boundary layer interaction which causes it to separate.

Clearly, additional resolution is needed to resolve the separation region. However, for the sake of simple CPU time comparisons with the adapted cases to evaluate the proposed scheme in Phase I, this resolution was deemed adequate. The computed solution is also somewhat different from that presented in Refs. (15) and (21). This is because different sorts of boundary conditions were used for the subsonic outflow of the boundary layer. Since substantial time and effort had already been invested in this fine grid solution, it was decided to retain the same characteristic subsonic outflow boundary condition for the adapted grid solutions presented in Subsections 4.4 and 4.5. A solution based upon Riemann invariant subsonic outflow boundary condition was only carried out for the fully adapted solution, and it is presented in Subsection 4.7. The latter agrees very well with the solutions in the cited references.

The CPU time needed to converge to the same level of accuracy as the Euler solution of Subsection 4.2 on the Data General machine was found to be 178 minutes. For this case the same minimum time steps were used for the viscous and inviscid contributions. When different time steps were used for these contributions, the total convergence time was 165 minutes. Thus it is found that using different time steps for viscous and inviscid contributions marginally accelerates the convergence process. Since the solutions for these

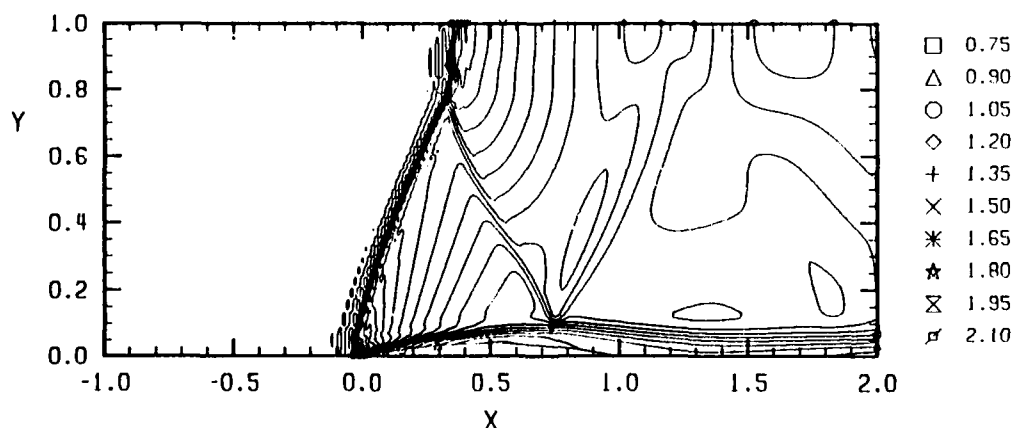


Figure 11. Density Contours for the Full Navier-Stokes Solution for 8% Circular Arc Cascade on a 51x173 Mesh System. The Increment for Density Contours is 0.05.

two cases were nearly identical, only the most restrictive time step solution is presented here. The CPU time for the full NS solution is about a factor of two longer than for the Euler solution. It is reasonable to assume that this ratio is Reynolds number dependent.

#### 4.4 Physics Adaptation Solution

Figure 12 shows the grids corresponding to the three blocks that were used for the physics adapted solution. The composite mesh is the same as in the previous two cases. The domain is split into Euler, Interaction and NS blocks and only part of the Euler block is shown. The viscous block is chosen so that the boundary layer will be captured. The interaction block will correspond to those parts of the boundary layer which may have been missed in the NS block, and the regions where the reflected shock wave interacts with the boundary layer.

Figure 13 shows the solution corresponding to physics adaptation procedure. Those cells in the interaction block that lie near the region of intersection of reflected shock wave and the boundary layer are automatically tagged as viscous cells. Except for some minor changes, the solution is essentially the same as the full NS solution.

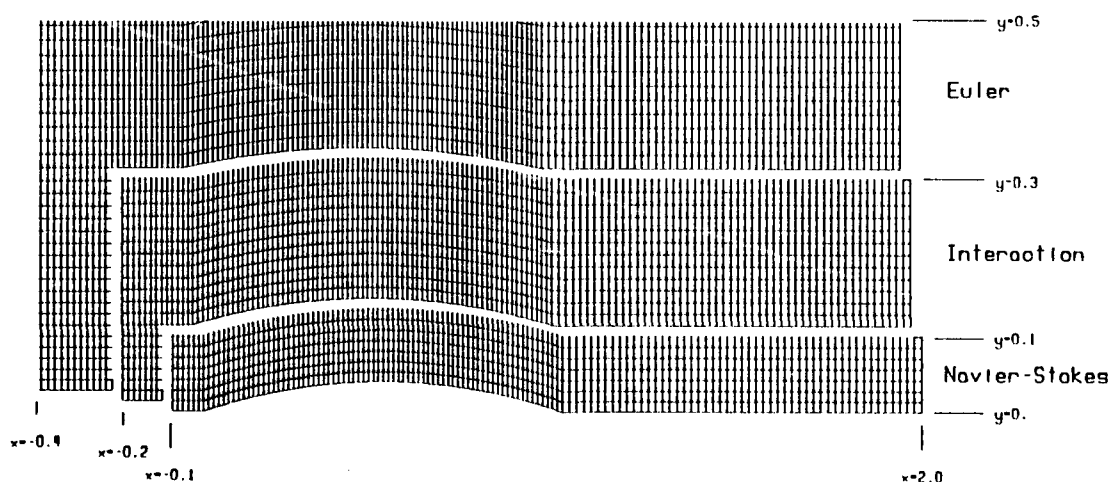


Figure 12. Domain Splitting for Physics Adaptation Solution for 8% Circular Arc Cascade on a Composite Mesh of 51X173 Points.

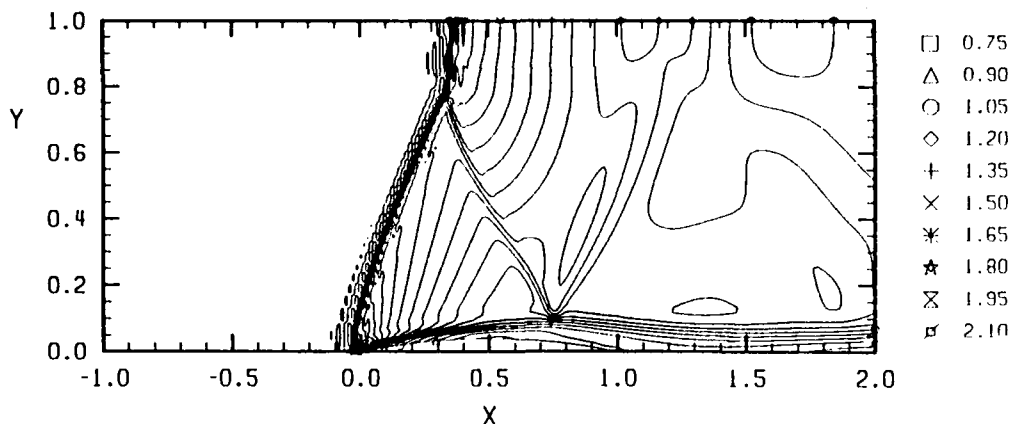


Figure 13. Density Contours for the Physics Adapted Solution for 8% Circular Arc Cascade on a 51X173 Initial Grid. Three Separate Blocks were Used for NS, Interaction and Euler Schemes. The Increment for Density Contours is 0.05.

The CPU time needed to converge to the same level of accuracy as the previous two solutions was found to be 82 minutes. The CPU time needed for this case is only 14% more than that required for the full Euler solution. This significant saving is achieved because the decision basis is limited only to the Interaction block and because different time steps are used for viscous and inviscid contributions.

#### 4.5 Physics and Subdomain Adaptation Solution

Figure 14a shows the spatially adapted grid for a coupled solution of physics adaptation and resolution adaptation. Two levels of spatial embedding are carried out for resolution adaptation. Note that the base grid is twice as coarse as the grid used for previous computations. Furthermore, the finest mesh is half as fine as the previous case. The domain splitting for the schemes is same as in the previous computation.

In order to avoid excessive numerical diffusion of the boundary layer on this initial coarse grid, the regions in the boundary layer were manually pre-embedded to a single level. This manual subdivision of the grid is accomplished as part of the solution of IBCG. The subdomain adaptation (both grid division and fusion) is carried out automatically after a specified

number of iterations, and as is apparent from the figure it can be achieved recursively. There are 11780 cells and 9820 nodes in the computational domain for the final grid. First differences of density and x-velocity have been used to accomplish spatial subdomain adaptation.

Figure 14b shows the solution corresponding to physics/subdomain adaptation procedure. Additional resolution due to subdomain adaptation allows the shock and boundary layer to be better resolved than the previous global solutions. The solution in the other regions is very nearly the same.

The CPU time needed to converge, to a level of accuracy of RMS errors being less than  $10^{-5}$ , was found to be 31 minutes. This means that the fully adapted solution requires only 17% of the CPU time compared to that of the full Navier-Stokes solution. However, even this comparison is not appropriate since the fully adapted solution is twice as fine as the full NS solution. This point is discussed further in the next subsection.

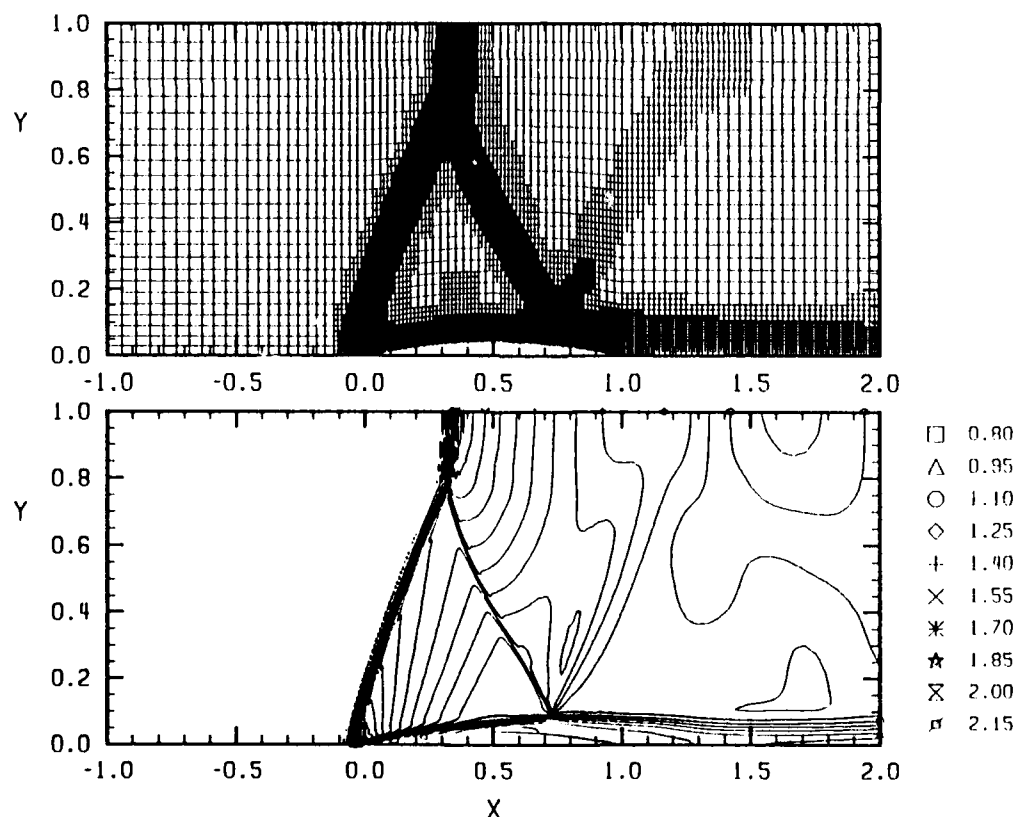


Figure 14. (a) Final Spatially Adapted Grid Obtained by Utilizing SCRAP Technique, and (b) Density Contours on this Grid with an Increment of 0.05.

#### 4.6 CPU Time Comparisons

Figure 15 summarizes the CPU time comparisons for various cases. The first column shows the actual CPU times in minutes for the various approaches. The second column shows the normalized computer times, where the Euler solution is regarded as the reference case. The third column indicates how fast a particular approach is in reference to the full NS solution. The normalized CPU time shows that full NS solution is 2.5 times slower than Euler solution, whereas the physics adapted solution for the same grid takes only 14% longer than the Euler equations to converge for the model problem.

It is seen that the fully adapted solution converges about 5.7 times faster than the full NS solution. However, since the finest resolution of the adapted grids is half as much as the global grid, a rough estimate shows that the full NS solution with the same fine grid will take about 8 times longer to converge. This estimate takes into account that there are four times as many number of cells with twice the number of time steps. Furthermore, physics adaptation itself is about 2.5 times faster. This translates into a total additional factor of 20 w.r.t. the full NS solution. Hence, the fully

METHOD	CPU TIME (min)	NORMALIZED CPU TIME	FACTOR
EULER	72	1.0	2.5
FULL NS	178	2.5	1.0
PHYSICS ADAPTATION	82	1.1	2.2
PHYSICS & SUBDOMAIN ADAPTATION	31	0.4	5.7*

\* 115 times faster than corresponding full NS solution.

Figure 15. CPU Time Comparisons for Various Approaches.

adapted solution is about two orders of magnitude faster than the corresponding full NS solution. With additional spatial levels of adaptation, one can achieve even larger factors. For 3-D computations, this saving will be even more substantial.

#### 4.7 Adapted Solution With Alternate Outflow Boundary Condition

In order to compare the solution of the viscous problem with those in Refs. (15) and (21), the fully adapted solution was carried out with similar outflow boundary conditions. The subsonic outflow boundary condition corresponds to the extrapolation of Riemann invariants from the interior domain. Pressure was inferred from the supersonic portion of the boundary layer and was held fixed for the subsonic part. Only the fully adapted solution was repeated because it runs significantly faster.

The initial composite grid was chosen to be composed of 25 x 25 points because the resolution along the x-axis is of less importance. Furthermore, the grid in the boundary layer was additionally compressed compared to the previous computations. The scheme-splitted blocks were similar to the previous cases.

Figure 16a shows the final adapted grids with two levels of spatial adaptation. There are a total of 3852 nodes and 4624 cells in this configuration.

Figure 16b shows the density contours for the grid in Fig. 16a. It is observed that the boundary layer is better resolved than any of the previous cases. This means that the incoming flow observes a somewhat smaller bump, and the flow is less choked. The Mach disc at the top surface is less severe, and the reflected shock wave reaches the boundary layer aft of the airfoil trailing edge. The boundary layer re-attaches itself at an earlier location, and the shock reflection emanating from the boundary layer is somewhat stronger.

Figure 17 presents the contours of Mach number. These agree very well with the computations presented in Refs. (15) and (21). Figure 18 represents the distribution of pressure along the airfoil surface; the agreement with Refs. (15) and (21) is again very good.



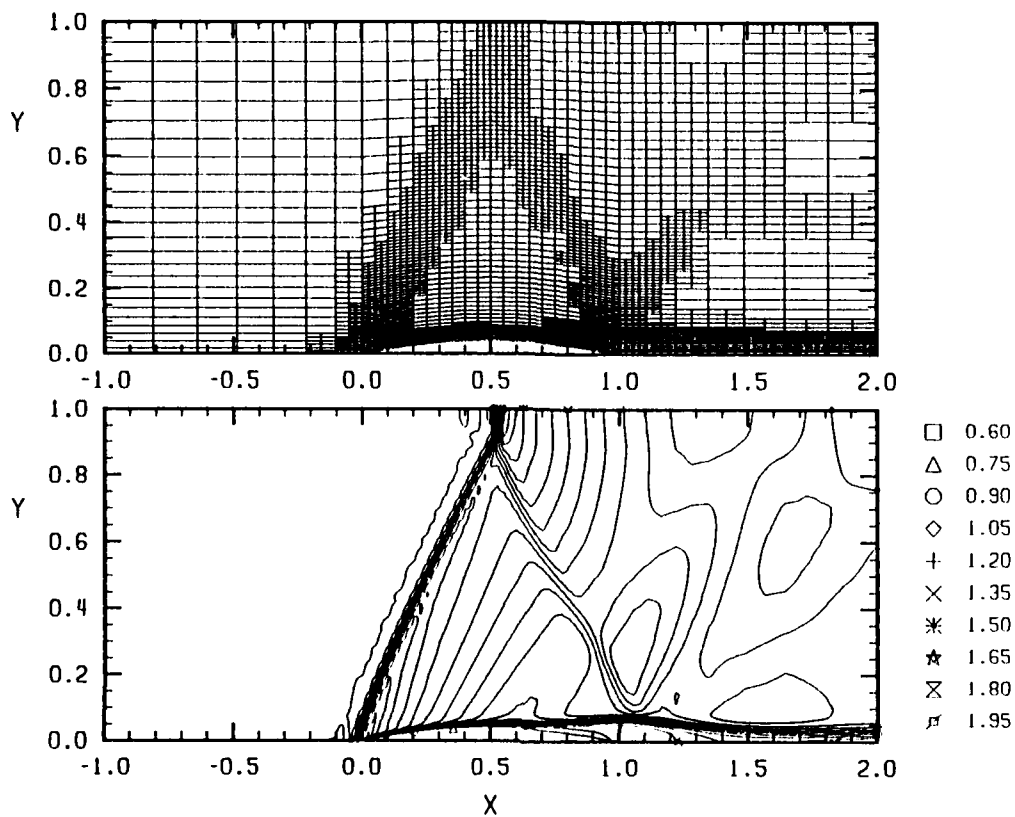


Figure 16. (a) Final Spatially Adapted Grid Obtained by Utilizing SCRAP Technique, and (b) Density Contours on this Grid with an Increment of 0.05.

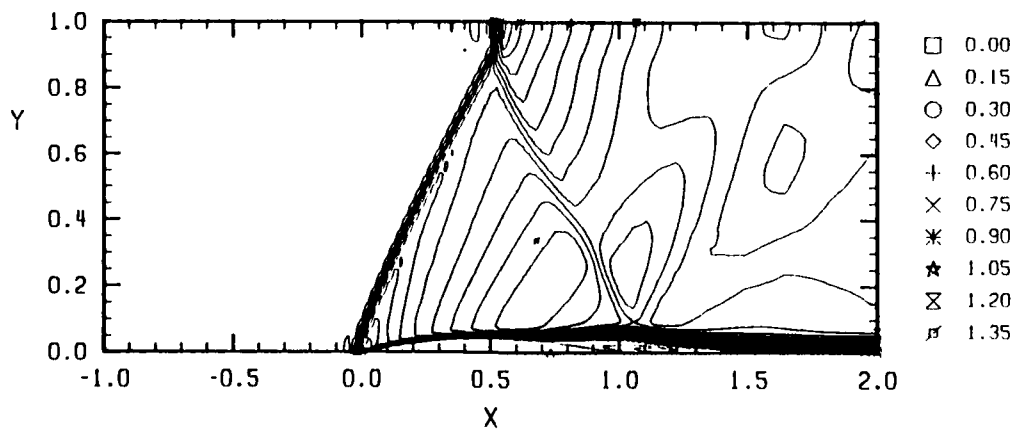


Figure 17. Mach Number Contours for the Fully Adapted Solution for an 8% Circular Arc Cascade. The Contour Increments are 0.05.

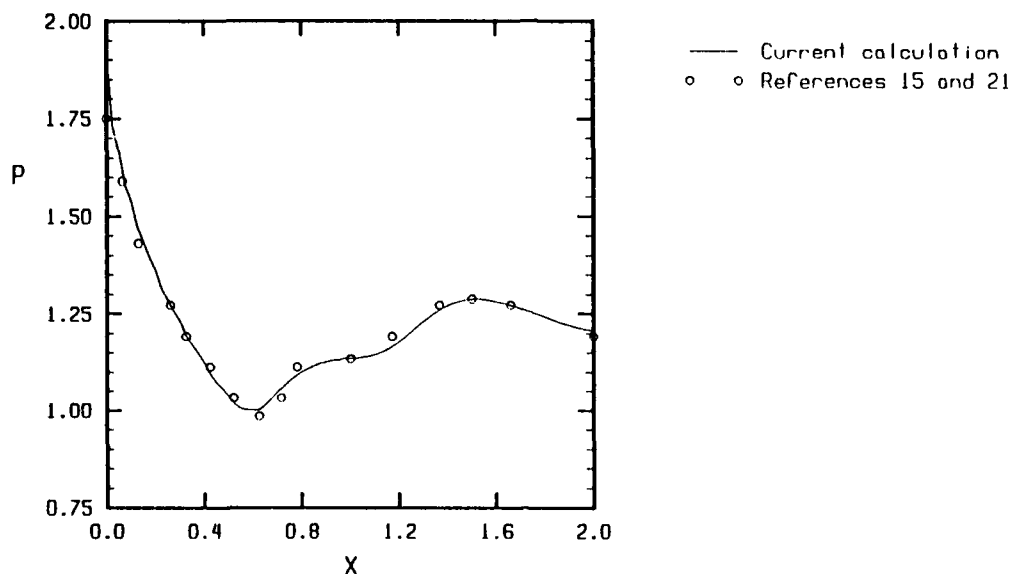


Figure 18. Pressure Distribution Along the Lower Channel Wall for the 8% Circular Arc Cascade.

The results presented in this section clearly demonstrate that simultaneous physics and resolution adaptation works for the sample problem. The technique provides significant savings in CPU time and storage over conventional structured grid algorithms employing a full NS solver. An increase of about two orders of magnitude in computational speed was achieved when compared to a conventional full NS solution over a fine grid of the same mesh size as the smallest grid size chosen by the spatial resolution adaptation.

## 5. CONCLUSIONS AND SUMMARY

The Phase I effort has successfully demonstrated the basic feasibility of simultaneous physics and subdomain adaptation for multiblock grids. The technique provides significant savings in CPU time and storage over conventional structured grid algorithms employing a full NS solver over a fine grid of the same mesh size as the smallest grid size chosen by the spatial resolution adaptation. The SCRAP technique yields essentially the same results as the global calculation over a fine grid system. The results were also compared with previous computations and were found to be in good agreement. This successful demonstration of proof of concept in Phase I has formed the basis for extension to 3-D in Phase II.

The results presented in Section 4 clearly demonstrate that simultaneous physics and resolution adaptation works for the sample problem. An increase of about two orders of magnitude in computational speed has been achieved when compared to a conventional full NS solution over fine grids. The savings in computer storage is also substantial since globally fine grids are not needed. The agreement of the full adapted solution with globally refined solutions and previous computations is excellent. The features that enhance the efficiency of the technique include (1) different solvers on different blocks and their dynamic switching, (2) different time steps for viscous and inviscid contributions, and (3) subdomain adaptation. Other advantages of the scheme include:

- ease in generating grids for complex geometries,
- ease in handling complex physics of specific zones,
- accurate modeling of features like shock waves, recirculation regions, etc.,
- greater speed compared to traditional schemes,
- greater accuracy compared to traditional schemes,
- comparatively less storage required for same accuracy.

The advantages for the SCRAP technique for three spatial dimensions will be even more dramatic. Formulation of the data management connectivities for 3-D will be carried out in Phase II. This work will produce a user-friendly software package that will alleviate the need for a single grid and allow concentration of computational resources in modeling complex physics only in regions of importance. This technique will result in an analysis tool that can be applied in carrying out flow simulations for complex and irregular geometries at large angles of attack. Such simulations often involve physically complex conditions of shock waves, boundary layers, separation and recirculation regions and their interactions. This analysis tool will more accurately predict the resultant forces and moments on practical aerodynamic geometries. This will lead to an accurate computation of the trajectory and design of guided projectiles.

## 6. REFERENCES

1. A. M. Agnone, V. Zakkay, and K. S. Parmar, "Experimental Investigation of the Hypersonic Flow Over a Six Finned Configuration," U. S. Army Research Office, Research Triangle Park, NC, ARO 20723.3-EG (June 1987).
2. J. Sahu and C. J. Nietubicz, "Numerical Flow Field Calculations for a Missile Configuration at  $M=.6$ ," U. S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD, Memorandum Report BRL-MR-3491 (January 1986).
3. W. B. Sturek, D. C. Mylin, and C. C. Bush, "Computational Parametric Study of the Aerodynamics of Spinning Bodies at Supersonic Speeds," U. S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD, ARBRL-TR-02358 (August 1981).
4. L. B. Schiff and W. B. Sturek, "Numerical Simulation of Steady Supersonic Flow Over an Ogive Cylinder Boattail Body," U. S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD, ARBRL-TR-02363 (September 1981).
5. W. B. Sturek and L. B. Schiff, "Computations of the Magnus Effect for Slender Bodies in Supersonic Flow," U. S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD, ARBRL-TR-02384 (December 1981).
6. P. Weinacht, B. J. Guidos, L. D. Kayser, and W. B. Sturek, "PNS Computations for Spinning and Fin-Stabilized Projectiles at Supersonic Velocities," U. S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD, Memorandum Report BRL-MR-3464 (September 1985).
7. P. Weinacht, B. J. Guidos, W. B. Sturek, and B. A. Hodes, "PNS Computations for Spinning Shell at Moderate Angles of Attack and for Long L/D Finned Projectiles," U. S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD, Memorandum Report BRL-MR-3522 (June 1986).
8. M. M. Pervaiz, "A Strategy Combining Regionally Adapted Processes (SCRAP)," Spectral Sciences, Inc. Rpt. No. SSI-4323. Proposal submitted to the Army Research Office (January 1990).
9. M. M. Pervaiz and J. R. Baron, "Temporal and Spatial Adaptive Algorithm for Reacting Flow," Communications in Applied Numerical Methods, 4, 1, Pages 97-111 (January 1988).
10. M. M. Pervaiz and J. R. Baron, "Spatio-Temporal Adaptation Algorithm for Two-Dimensional Reacting Flows," AIAA Paper 88-0510 (January 1988).

11. M. M. Pervaiz, "Spatiotemporal Adaptive Algorithm for Reacting Flows," CFDL-TR-88-5, Ph.D. Thesis, Vol. I, Dept. of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA (May 1988).
12. M. M. Pervaiz and J. R. Baron, "Temporal and Spatial Adaptive Algorithm for Reacting Flows," AIAA Journal, 27, 10, pp. 1368-1376 (October 1989).
13. R. H. Ni, "A Multiple Grid Scheme for Solving the Euler Equations," AIAA Journal, 20, 11 pp. 114-121 (November 1982).
14. R. Peyret and T. Taylor, Computational Methods for Fluid Flow, Springer-Verlag, Inc. (1983).
15. I. Kallinderis, "Adaptation Methods for Viscous Flow," CFDL-TR-89-5, Ph.D. Thesis, Dept. of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA (May 1989).
16. D. Anderson, J. Tannehill, and R. Pletcher, Computational Fluid Mechanics and Heat Transfer, McGraw-Hill (1984).
17. R. D. Richtmyer and K. W. Morton, Difference Methods for Initial Value Problems, Wiley and Sons, New York (1967).
18. K. Rajagopal, W. Lick, K. Szema, S. Ramakrishnan, and C. Chen, "A Versatile Multi-Zone Gridding Technique for Complex Geometries," AIAA Paper 90-0011 (January 1990).
19. J. Dannenhoffer and J. Baron, "Grid Adaptation for the 2-D Euler Equations," AIAA Paper 85-0495 (January 1985).
20. R. A. Shapiro and E. M. Murman, "Cartesian Grid Finite Element Solutions to the Euler Equations," AIAA Paper 87-0559 (January 1987).
21. J. Kallinderis and J. R. Baron, "Adaptation Methods for a New Navier-Stokes Algorithm," AIAA Journal, 27, 1, pp. 37-43 (January 1989).